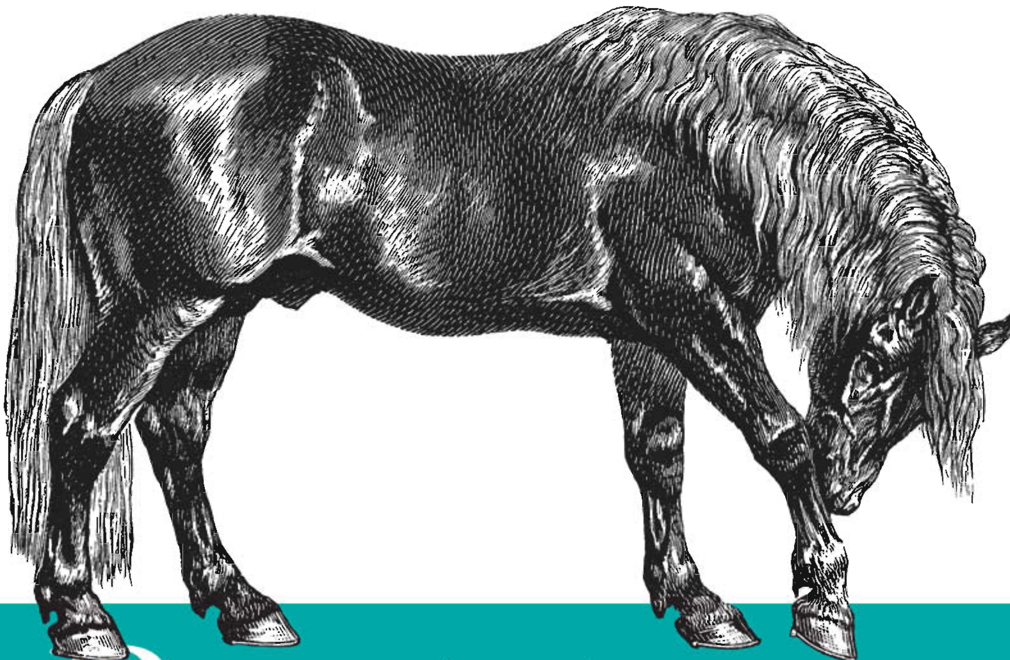
TURING 图灵程序设计丛书

*Bootstrap*
*Responsive Web Development*



# Bootstrap 用户手册

## 设计响应式网站


[美] *Jake Spurlock* 著
*Dave Winer* 序
李松峰 译



O'REILLY®

人民邮电出版社
POSTS & TELECOM PRESS

# 数字版权声明

图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

图灵程序设计丛书

# Bootstrap用户手册

## Bootstrap

[美] Jake Spurlock 著
李松峰 译

O'REILLY®

*Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo*

O'Reilly Media, Inc.授权人民邮电出版社出版

人民邮电出版社
北京

图书在版编目（CIP）数据

Bootstrap用户手册 : 设计响应式网站 / （美）斯珀洛克（Spurlock,J.）著 ; 李松峰译. -- 北京 : 人民邮电出版社, 2013.9
（图灵程序设计丛书）
书名原文：Bootstrap
ISBN 978-7-115-32735-2

Ⅰ. ①B… Ⅱ. ①斯… ②李… Ⅲ. ①网页制作工具－手册 Ⅳ. ①TP393.092-62

中国版本图书馆CIP数据核字(2013)第187388号

## 内 容 提 要

Bootstrap 是 Mark Otto 和 Jacob Thornton 在 Twitter 公司共同开发的一个网站开发框架，2011 年 8 月发布至今已经成为 Github 上最受关注的开源项目。从最初 CSS 驱动的项目到目前内置很多 JavaScript 插件和图标，并且涵盖表单和按钮元素、支持响应式 Web 设计，Bootstrap 为 Web 开发提供了前瞻性的设计和开发基础，能够大幅提升开发效率，已被很多流行网站采用。本书全面介绍了 Bootstrap 为前端开发提供的所有工具，包括网格布局系统、预定义 CSS 样式、内置界面组件和交互式 JavaScript 插件。

本书内容系统、结构完整、讲解简明、方便实用，是前端开发人员使用 Bootstrap 的最佳参考书，适合所有前端开发人员和希望了解 Bootstrap 的读者阅读参考。

◆ 著　　[美] Jake Spurlock
译　　李松峰
责任编辑　刘美英
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京　　　　印刷

◆ 开本：880×1230　1/32
印张：4.375
字数：118千字　　　　2013年 9 月第 1 版
印数：1－4 000册　　　　2013年 9 月北京第 1次印刷
著作权合同登记号　图字：01-2013-5238号

定价：35.00元
**读者服务热线：(010)51095186转604　印装质量热线：(010)67129223**
**反盗版热线：(010)67171154**
**广告经营许可证：京崇工商广字第 0021 号**

# 版权声明

©2013 by Jake Spurlock.

Simplified Chinese Edition,jointly published by O'Reilly Media,Inc.and Posts & Telecom Press, 2013 Authorized translation of the English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2013。

简体中文版由人民邮电出版社出版，2013。英文原版的翻译得到 O'Reilly Media, Inc. 的授权。此简体中文版的出版和销售得到出版权和销售权的所有者 —— O'Reilly Media, Inc. 的许可。

版权所有，未得书面许可，本书的任何部分和全部不得以任何形式重制。

# O’Reilly Media, Inc.介绍

O’Reilly Media 通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自 1978 年开始，O’Reilly 一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O’Reilly 的发展充满了对创新的倡导、创造和发扬光大。

O’Reilly 为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了 Make 杂志，从而成为 DIY 革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O’Reilly 的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O’Reilly 现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项 O’Reilly 的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

## 业界评论

“O’Reilly Radar 博客有口皆碑。”

——Wired

“O’Reilly 凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O’Reilly Conference 是聚集关键思想领袖的绝对典范。”

——CRN

“一本 O’Reilly 的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim 是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照 Yogi Berra 的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去，Tim 似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

# 目录

# 序

我在 Mac 面世之前就开发软件了，因此我记得当时技术界的反应。很大程度上，技术圈的人都对此不太感冒。有意思的是，今天很多人对 Bootstrap 的负面影响议论纷纷，像极了 1984 年人们对 Mac 的指指点点。但两个事件中一致的是，人们不喜欢的东西，对他们却非常重要。

苹果意识到有些事所有软件都必须做，那为什么这些软件不采用相同的方法来做呢？如果能做到这一点，开发和调试就容易多了，而更重要的是，用起来也就容易多了！如果只有一种创建菜单的方式，那么用户只要在一个应用上学会使用菜单，就会使用任何应用中的菜单了。滚动条、窗口、键盘、鼠标、打印机、音乐，不都一样吗？

那时，程序员（包括我）之所以不喜欢 Mac，是因为觉得它要拿我们做过的东西转变成商品去卖钱。另外，这种大一统的方式存在局限性。有一些应用的界面元素并不规范。怎么办？好，你自己来改，只能这样。

这在技术上就叫重构。如果你发现自己一遍一遍地在重复做一件事，不妨再努力一次做得更好一点，做出个 API 来，保证它灵活好用。然后，就再也不用重复了。这正是软件也能写得像摩天大楼那样高大复杂的奥秘。五年前最前沿的技术如今已经融入了操作系统。这就是进步，这样多好啊！

同样的情形在网站开发过程也随处可见。事实上，以 Mac 当年对未来

的愿景观之，Web 开发领域的重复劳动以及开发环境的落后程度是令人沮丧的。然而，Bootstrap 至少担负起了重构的责任。如果我想做个菜单，那就用它来做好了。没错，我的菜单跟别人的没什么区别。但这对用户是好事儿啊。用户不用再学习第二种菜单、第三种菜单的用法了。

Bootstrap 是必需的、绝对必要的，它的迅速普及就是明证。不光我的服务器端软件会用到它，我开发的模板语言里也少不了它，这样只要用户想使用 Bootstrap，随时随地用就可以了。什么也不用做，也不用包含什么库。就像服务器硬件的一部分，跟苹果 1984 年在 Mac OS 上所做的一样。

跟所有重要的技术一样，Bootstrap 只是“够好”而已，并非完美。换句话说，Mark Otto 和 Jacob Thornton 还可以让自己的重构更上一层楼。然而，假如他们不断追求让代码更简洁、外观更漂亮，那恐怕就没有那么好用了。Bootstrap 为你我这样忙于写网站的人彻底解决了用户界面的问题。

不过，我认为这只是开始。我预感到 Bootstrap 会成为 Web 的一个组件，或者说它很可能出现在你会用到的所有开发工具中。有朝一日——应该不会太久，设计人员将能够不修改任何样式和脚本，就给整个网站换一次皮肤。我没发现使用 Bootstrap 会带来什么局限性。Bootstrap 不能取代设计师，相反，它是设计师手中的利器，也拓展了设计师的想象空间。

今天，能在 O'Reilly 出一本书往往标志着一种新技术走向成熟。现在，Bootstrap 的书也出版了。作者 Jake Spurlock 是一位 Web 开发人员，已经使用 Bootstrap 写了很多网站，并在技术大会上分享过他的经验。我很荣幸，据说他是在看了我写的一系列博客文章之后才决定使用 Bootstrap 的。现在，我非常高兴地把他的书推荐给各位读者。来听听他为大家讲解 Bootstrap 的魔法吧，让自己也拥有推动世界进步的魔力！

Dave Winer

Scripting News 编辑

2013 年 1 月于纽约

# 前言

Bootstrap 是一个用于构建响应式网站的前端框架。无论你想构建应用程序、博客还是 CMS 网站，Bootstrap 都完美适用，只要你想得到，它都能行。Bootstrap 把 HTML、CSS 和 JavaScript 组合起来，为构建稳定的网站提供了基础设施，也能提高开发效率。基于默认的网格系统，布局变成了小菜一碟，按钮、导航和表格也都活脱脱地赏心悦目。十几个 JavaScript 插件更为我们赋予网站交互能力提供了强大保障。

## 读者对象

本书读者应该对 HTML、CSS 和 JavaScript 都比较熟悉，而且对构建响应式网站、使用 Bootstrap 的 JavaScript 插件，以及使用流行的开源框架创建网站充满好奇心。

## 不适合谁

本书不适合能看懂 Bootstrap 文档的读者。与很多人一样，我也是从在线文档学起的，当时使用的是 Bootstrap 1.3，后来就升级到 Bootstrap 1.4。然后，我又基于 Bootstrap 2.0 建了一个大网站。如果你喜欢编写语义化的 HTML，使用 Bootstrap 没错。

## 本书宗旨

如果你有编写 HTML、CSS 和 JavaScript 的经验，本书可以作为你编写灵活代码和响应式网站的参考。当然，相关的概念和语法都很简单，因为本书也遵循语义 HTML 和 CSS 的风格。

## 本书内容

本书讨论了使用 Bootstrap 构建网站，从基本的项目和文件结构开始，然后是网格系统和布局方式，再后来讨论 HTML 元素特别是表单、表格、按钮等界面元素。在对 Bootstrap 有了基本的理解之后，我们会讨论漂亮的导航条、面包屑式导航、媒体对象等。接着 JavaScript 插件登场，比如下拉菜单、传送带、模态框，它们为网站提供了交互功能。

## 为什么写这本书

我不是 Bootstrap 专家，把自己当成专家可不得了，写成这样就不行了。

我就是一个普通的开发人员，通过 Dave Winer 的博客知道了 Bootstrap。我觉得不错，就在新网站中使用了 Bootstrap。我感觉有必要跟大家分享自己使用 Bootstrap 的心得。希望我从一个初学者角度写的这本书，能够帮更多人认识到 Bootstrap 的价值，希望自己的经历对大家有用，也请专家们批评指正。

## 其他资源

看书或许并不是学习 Bootstrap 的唯一方式。这取决你想学什么，或者为什么学。

如果你只想初步了解 Bootstrap，那么其在线文档已经足够了。Jacob Thornton 和 Mark Otto 为此提供了很多精美的示例、HTML 片段及其他对你入门有帮助的信息。Bootstrap 的在线文档写得很好，本书结构也参考了该文档。

如果你想对这个开源项目做点贡献，可以在 GitHub 上提交请求或使用跟踪程序，获得项目的更新、下载、文档。

## Bootstrap适合你吗

如果你只想找 JavaScript 插件或 CSS 重置文件，Bootstrap 可能就太大了。如果你只是不喜欢 Bootstrap 的默认界面元素，那不要紧，因为你可以轻易定制这些元素的外观，而且去掉相关的标签也易如反掌。如果你想快速构建响应式网站，Bootstrap 绝对是明确的选择。

## 排版约定

本书使用的排版约定如下。

- 楷体
  表示新的术语。

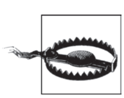
- 等宽字体
  表示程序片段，也用于在正文中表示程序中使用的变量、函数名、命令行代码、环境变量、语句和关键词等代码文本。

- 加粗的等宽字体
  表示应该由用户逐字输入的命令或者其他文本。

- 倾斜的等宽字体
  表示应该由用户输入的值或由上下文决定的值替换的文本。



这个图标代表小窍门、建议或说明。



这个图标代表警告信息。

# 使用代码

本书就是要帮读者解决实际问题的。也许你需要在自己的程序或文档中用到本书中的代码。除非大段大段地使用，否则不必与我们联系取得授权。因此，用本书中的几段代码写成一个程序不用向我们申请许可。但是销售或者分发 O'Reilly 图书随附的代码光盘则必须事先获得授权。引用书中的代码来回答问题也无需我们授权。将大段的示例代码整合到你自己的产品文档中则必须经过许可。

使用我们的代码时，希望你能标明它的出处。出处一般要包含书名、作者、出版商和 ISBN，例如：*Bootstrap* by Jake Spurlock (O'Reilly). Copyright 2013 Jake Spurlock, 978-1-449-34391-0。

如果还有其他使用代码的情形需要与我们沟通，可以随时与我们联系：permissions@oreilly.com。

# 欢迎指正

希望读者可以通过本书学到想学的东西，同时也希望你能帮助别人学习 Bootstrap。帮助别人的方式有很多种。

- 发现本书有技术问题或者解释不清，或其他可以改进的地方，请提交勘误。
- 如果你喜欢（或不喜欢）这本书，请写出书评。可以直接在 Amazon.com 或 O'Reilly 网站本书页面上留言。指出哪些地方讲得清楚，哪些地方还不够清楚，对其他读者和我都将有很大帮助。
- 如果你想分享学习 Bootstrap 的心得，请付诸行动，你可以写博客、发微博、写书、搞培训，怎么都行。

我会尽力根据勘误和评论更新这本书。就算书已经印刷上市，我也会在新版本中更正。要是你购买了电子版，可以免费获得新版本。我觉得这本书不会升级太快，除非 Bootstrap 有很大变化。但愿这本书能让读者朋友有所收获，激发你分享的兴趣。

# Safari® Books Online

Safari Books Online（www.safaribooksonline.com）是应需而变的数字图书馆。它同时以图书和视频的形式出版世界顶级技术和商务作家的专业作品。

Safari Books Online 是技术专家、软件开发人员、Web 设计师、商务人士和创意人士开展调研、解决问题、学习和认证培训的第一手资料。

对于组织团体、政府机构和个人，Safari Books Online 提供各种产品组合和灵活的定价策略。用户可通过一个功能完备的数据库检索系统访问 O'Reilly Media、Prentice Hall Professional、Addison-Wesley Professional、Microsoft Press、Sams、Que、Peachpit Press、Focal Press、Cisco Press、John Wiley & Sons、Syngress、Morgan Kaufmann、IBM Redbooks、Packt、Adobe Press、FT Press、Apress、Manning、New Riders、McGraw-Hill、Jones & Bartlett、Course Technology 以及其他几十家出版社的上千种图书、培训视频和正式出版之前的书稿。要了解 Safari Books Online 的更多信息，我们网上见。

# 联系我们

请把对本书的评价和问题发给出版社。

美国：

O'Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938（从美国和加拿大拨打）
707-829-0515（在本地或从其他国家拨打）
707-829-0104（传真）

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室（100035）
奥莱利技术咨询（北京）有限公司

O'Reilly 的每一本书都有专属网页，你可以在那儿找到本书的相关信息，包括勘误表、示例代码以及其他信息。本书的网站地址是：

http://shop.oreilly.com/product/0636920027867.do

中文版地址：

http://www.oreilly.com.cn/index.php?func=book&isbn=978-7-115-32735-2

对于本书的评论和技术性问题，请发送电子邮件到：

bookquestions@oreilly.com

要了解更多 O'Reilly 图书、培训课程、会议和新闻的信息，请访问以下网站：

http://www.oreilly.com

我们在 Facebook 的地址如下：

http://facebook.com/oreilly

请关注我们的 Twitter 动态：

http://twitter.com/oreillymedia

我们的 YouTube 视频地址如下：

http://www.youtube.com/oreillymedia

# 致谢

首先感谢 Dave Winer 引领我使用 Bootstrap，然后感谢 Simon St. Laurent 为我提供了写书的机会。好朋友 Roseanne Fallin 和 Tony Quartorolo 的反馈让这本书成为可能。我希望这本书能帮读者走上正轨。还要感谢 Melissa Morgan 支持我在 MAKE 似乎有点冒险的开发思路。

特别要感谢我的妻子 Melissa，感谢她的坚忍和对我的鼓励。感谢我的儿子 Rush，他居然理解我需要“工作”，还要感谢我女儿 Hailey 那暖暖的微笑和紧紧的拥抱。我爱我家，感谢她们对我的支持。

第 1 章

# Bootstrap提供的网站框架

## 1.1 Bootstrap到底是什么

Bootstrap 是 Mark Otto 和 Jacob Thornton 共同开发的一个开源框架。最初发布 Bootstrap 的时候，他们两位都在 Twitter 工作。当时，他们面临着为公司内部提供一套标准化的前端开发工具的需求。在发布 Bootstrap 的那篇博客中，Mark Otto 这样介绍他们的新项目：

> Twitter诞生初期，为了完成前端开发任务，工程师们几乎尝试了自己熟悉的所有库。应用之间的不一致性导致它们难以扩展和维护。Bootstrap最初就是这个问题的一个解决方案，而且在Twitter内部第一次Hackweek上得到了认可。Hackweek结束后，我们手头上的项目已经相当稳定，可以在公司内部部署使用了。
>
> ——Mark Otto
> https://dev.twitter.com

Bootstrap 是 2011 年 8 月发布的，发布之后就迅速走红。而且，它也从最初 CSS 驱动的项目，发展到内置了很多 JavaScript 插件和图标，并且涵盖了表单和按钮元素。Bootstrap 本身支持响应式 Web 设计，而且拥有一个非常稳健的 12 列、940 像素宽的网格布局系统。值得一提的是，Bootstrap 网站（http://getbootstrap.com）上还提供了一个构建工具，让你根据自己的需求选择 CSS 和 JavaScript 功能。所有这一切让前端 Web 开发有了前瞻性的设计和开发基础，开发效率倍增。上手使用 Bootstrap 非常简单，跟在网站中整合 CSS 和 JavaScript 没有什么区别。

对于新项目，Bootstrap 则提供了一整套有用的基本要素。一般来说，我在着手开发新项目的时候，都会整合 Eric Meyer 的 Reset CSS（http://meyerweb.com/eric/tools/css/reset/）等一些工具。自从有了 Bootstrap，只要在网站中引入 bootstrap.css 这个 CSS 文件，或者再加上 bootstrap.js 这个 JavaScript 文件，整个项目的基础平台就搭建好了。

## 1.2　Bootstrap的文件结构

```
bootstrap/
        ├── css/
        │   ├── bootstrap.css
        │   ├── bootstrap.min.css
        ├── js/
        │   ├── bootstrap.js
        │   ├── bootstrap.min.js
        ├── img/
        │   ├── glyphicons-halflings.png
        │   ├── glyphicons-halflings-white.png
        └── README.md
```

下载之后的 Bootstrap 包含三个文件夹：css、js 和 img。为简单起见，请大家把这三个文件夹放到项目根目录中。CSS 和 JavaScript 文件分别有一个压缩版，可以根据需要选择一个版本，没必要在项目中同时包含两个版本。我一般开发时使用未压缩版，而在发布时使用压缩版本。

## 1.3　基本的HTML模板

一般 Web 项目的 HTML 文件如下所示：

```
<!DOCTYPE html>
<html>
        <head>
                <title>Bootstrap 101 Template</title>
        </head>
        <body>
                <h1>Hello, world!</h1>
        </body>
</html>
```

使用 Bootstrap 时，则要包含它的 CSS 样式表和 JavaScript 文件：

```
<!DOCTYPE html>
<html>
        <head>
                <title>Bootstrap 101 Template</title>
                <link href="css/bootstrap.min.css" rel="stylesheet">
        </head>
```

```
        <body>
                <h1>Hello, world!</h1>
                <script src="js/bootstrap.min.js"></script>
        </body>
</html>
```

> 别忘了 HTML5 的文档类型声明。
>
> 只要包含 <!DOCTYPE html>，所有现代浏览器都启动标准模式。

## 1.4 全局样式

Bootstrap 预定义了一些样式。Bootstrap 1.0 使用的是以前的重置样式表，但 Bootstrap 2.0 则使用 Nicolas Gallagher 写的 Normalize.css（http://necolas.github.com/normalize.css/），HTML5 Boilerplate（http://html5boilerplate.com/）使用的也是这个样式表。这个样式表包含在 bootstrap.css 中。

特别值得注意以下几种默认样式，它们专门针对排版和链接：

- 从 body 中去掉了 margin，这样内容就会紧贴浏览器窗口边缘；
- 为 body 应用了 background-color: white;；
- Bootstrap 以 @baseFontFamily、@baseFontSize 和 @baseLineHeight 属性作为排版的基准，确保整个网站的标题及其他内容的行高一致；
- Bootstrap 通过 @linkColor 设置全局链接颜色，只在 :hover 状态下给链接添加下划线。

> 记住，要修改默认颜色，可以直接修改相应 .less 文件中的全局变量值。要么修改 scaffolding.less 文件，要么修改你自己的样式表。

## 1.5 默认网格系统

Bootstrap 默认的网格布局（图 1-1）包含 12 列，940 像素宽，不支持响应式布局。加载响应式 CSS 文件后，网格布局会根据视口

（viewport）宽度在 724 像素到 1170 像素之间伸缩。视口宽度小于 767 像素时，说明是平板电脑或更小的设备，布局中的列会垂直堆叠起来。默认宽度下，每列宽 60 像素，且向左平移 20 像素。图 1-1 展示了 12 列时的情况。

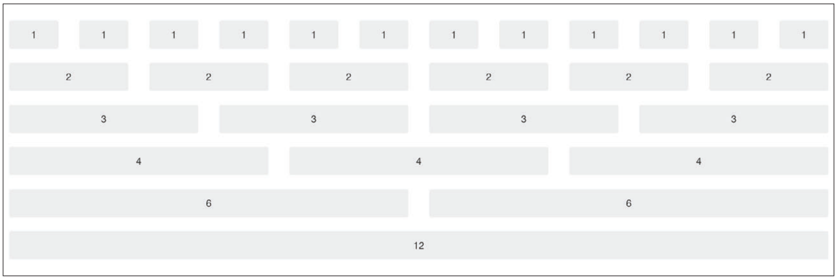



图 1-1：默认的网格

## 1.5.1　基本网格的HTML

创建简单布局时，给作为行的 <div> 添加 .row 类，给作为列的 <div> 添加表示横跨多少列的 .span* 类即可。因为网格布局可以包含 12 列，所以 .span* 中的 * 加起来必须等于 12。这样，可以设计出 3-6-3、4-8、3-5-4、2-8-2……布局，你懂了吧？

下面的代码使用了 .span8 和 .span4，总共横跨 12 列：

```
<div class="row">
  <div class="span8">...</div>
  <div class="span4">...</div>
</div>
```

## 1.5.2　平移列

使用 .offset* 类可以向右平移列，* 用于指定平移的列数。比如，下面代码中的 .offset2 会导致 .span7 向右平移 2 列（见图 1-2）：

```
<div class="row">
  <div class="span2">...</div>
  <div class="span7 offset2">...</div>
</div>
```

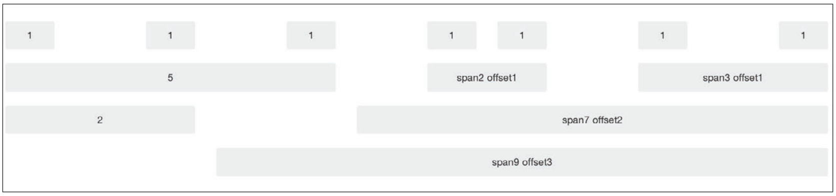



图 1-2：平移列

### 1.5.3　嵌套列

要嵌套列，只要在相应的 .span* 中再增加一个 .row，然后在其中包含与父容器列数相等的 .span* 即可（见图 1-3）：

```
<div class="row">
  <div class="span9">
    Level 1 of column
    <div class="row">
      <div class="span6">Level 2</div>
      <div class="span3">Level 2</div>
    </div>
  </div>
</div>
```

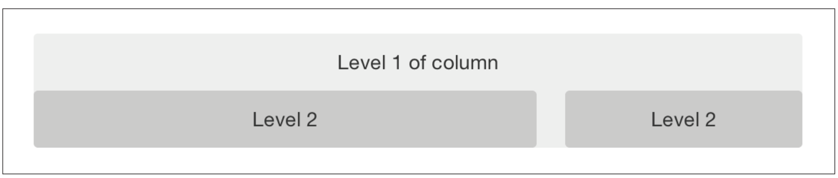



图 1-3：嵌套列

## 1.6　流式网格系统

流式网格系统的列宽定义使用百分比，而不是像素。流式网格系统与固定网格系统一样都具有响应能力，可适应不同的屏幕和设备。只要把某一行的 .row 改为 .row-fluid，这一行就会成为流式的。不改变列的类，是为了简化固定和流式网格的切换。想平移列？跟固定网格中一样，在要平移的列中添加 .offset* 类即可：

```
<div class="row-fluid">
  <div class="span4">...</div>
  <div class="span8">...</div>
</div>

<div class="row-fluid">
  <div class="span4">...</div>
  <div class="span4 offset2">...</div>
</div>
```

流式网格中的嵌套有点不一样。原因是在使用百分比的情况下，每个 .row 都会把列数重置为 12。举个例子，要想在一个 .span8 中嵌套两个等宽的列，不能使用两个 .span4（尽管两个 4 平分 8），而应该使用两个类为 .span6 的 <div>（见图 1-4）。这是为保证内容具有响应性，因为我们希望内容 100% 填满容器：

```
<div class="row-fluid">
  <div class="span8">
              <div class="row">
                      <div class="span6">...</div>
                      <div class="span6">...</div>
              </div>
  </div>
</div>
```

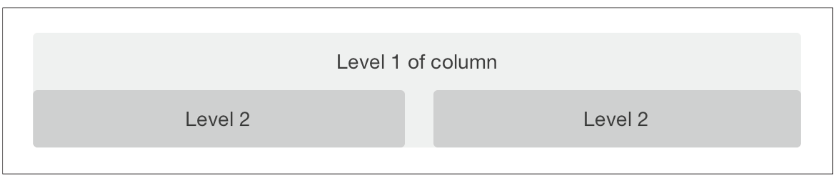



图 1-4：嵌套的流式网格

## 1.7 容器布局

要在页面中实现固定宽度、居中的布局，只要把内容统统放到 <div class="container">...</div> 中即可。如果既想实现流式布局，又想把所有内容都封装到一个容器里，可以使用 <div class="container-fluid">...</div>。流式布局适合很多应用、管理界面及其他项目。

# 1.8 响应式设计

要让 Bootstrap 支持响应式布局，必须在 <head> 标签中添加一个 <meta> 标签。另外，如果你下载的文件中没有响应式 CSS 文件，也要单独下载。响应式布局必需的标签和文件如下所示：

```
<!DOCTYPE html>
<html>
        <head>
                <title>My amazing Bootstrap site!</title>
                <meta name="viewport" content="width=device-width,
              initial-scale=1.0">
                <link href="/css/bootstrap.css" rel="stylesheet">
                <link href="/css/bootstrap-responsive.css" rel="stylesheet">
        </head>
```

如果发现 Bootstrap 没有响应能力，请确保这些标签一个也不少。目前，响应能力还不是默认就有的，因为不是所有网站都需要响应式布局。Bootstrap 的作者选择了让开发人员需要时添加，而不是不需要时删除响应功能。

## 什么是响应式设计

响应式设计是针对浏览设备优化页面中既有内容的一种方法。比如，在桌面浏览器中既可以看到网站的常规版本，也可以在用户接入更大的显示器时看到针对宽屏的布局。在平板电脑中看到的是针对其横屏和竖屏模式优化之后的布局。而在手机上，则是能够适应更窄宽度的布局。为了适用不同的屏幕宽度，Bootstrap 使用 CSS 的媒体查询（media query）来检测浏览器视口的宽度，然后再根据条件加载和应用调整布局的样式表。根据浏览器视口的宽度，Bootstrap 可以按照纵横比或宽度的范围来优化布局，但最主要还是使用 `min-width` 和 `max-width` 属性。

Bootstrap 内部支持 5 种不同的布局，都依赖于 CSS 媒体查询。最大布局中每一列的宽度为 70 像素，而常规布局中每一列宽度是 60 像素。在适合平板电脑的布局中，列宽缩小为 42 像素，再窄一点，每

一列就会流动起来，变成在垂直方向上堆叠，并且都与视口同宽（见表 1-1）。

表1-1　响应式媒体查询

| 目标设备 | 布局宽度 | 列　宽 | 列间距 |
|---|---|---|---|
| 大屏幕 | 至少 1200px | 70px | 30px |
| 常规 | 至少 980px | 60px | 20px |
| 平板竖屏 | 至少 768px | 42px | 20px |
| 手机到平板 | 至多 767px | 流式列，宽度不固定 | |
| 手机 | 至多 480px | 流式列，宽度不固定 | |

要根据媒体查询添加自定义的 CSS，可以像下面这样把所有规则写到一个 CSS 文件中，也可以单写一个 CSS 文件：

```
/* 大屏幕 */
@media (min-width: 1200px) { ... }

/* 平板竖屏到横屏到常规屏幕 */
@media (min-width: 768px) and (max-width: 979px) { ... }

/* 手机横屏到平板竖屏 */
@media (max-width: 767px) { ... }

/* 手机横屏及更小屏幕 */
@media (max-width: 480px) { ... }
```

对于大型网站，应该把不同的媒体查询分别保存在单独的 CSS 文件中。而在 HTML 的 <link> 标签中，可以根据条件加载它们。这样可以保证加载的文件相对小，但在响应式布局中会增加 HTTP 请示。如果你使用 LESS 编译 CSS，可以把它们都放到一个文件中：

```
<link rel="stylesheet" href="base.css" />
<link rel="stylesheet" media="(min-width: 1200px)" href="large.css" />
<link rel="stylesheet" media="(min-width: 768px) and (max-width: 979px)"
      href="tablet.css" />
<link rel="stylesheet" media="(max-width: 767px)" href="tablet.css" />
<link rel="stylesheet" media="(max-width: 480px)" href="phone.css" />
```

## 辅助类

Bootstrap 为响应式开发提供了一些辅助类（见表 1-2），但这些类还是

能不用就不要用。什么情况下用呢？比如在某些布局中显示或隐藏自定义元素。举个例子，常规布局中可以包含一个页眉，但在移动布局中需要简化它，只显示少量元素。此时，可以给页眉中需要隐藏的元素添加 `.hidden-phone` 类。

表1-2　媒体查询辅助类

| 类 | 手机 | 平板 | 电脑 |
| --- | --- | --- | --- |
| `.visible-phone` | 显示 | 隐藏 | 隐藏 |
| `.visible-tablet` | 隐藏 | 显示 | 隐藏 |
| `.visible-desktop` | 隐藏 | 隐藏 | 显示 |
| `.hidden-phone` | 隐藏 | 显示 | 显示 |
| `.hidden-tablet` | 显示 | 隐藏 | 显示 |
| `.hidden-desktop` | 显示 | 显示 | 隐藏 |

响应式开发有两种主要的思路。当前备受推崇的方法是从移动版本做起，最后才是桌面网站。Bootstrap 的思路刚好相反，你可以用它来搭建一个“差不多”的桌面网站。

不过，就算你想找的是移动开发框架，Bootstrap 仍然是值得考虑的。

# 第2章

# Bootstrap预定义的CSS样式

Bootstrap 提供了一组基本的具有预定义样式的 HTML 元素，用户通过类和自定义样式很容易增强这些元素。

# 2.1　排版

对页面排版，Bootstrap 使用的默认字体栈为 `"Helvetica Neue", Helvetica, Arial, sans-serif;`。这些字体也是主流操作系统默认支持的标准字体。万一明确指定的字体都不存在，那么负责兜底儿的 `sans-serif` 会告诉浏览器使用默认的字体。页面正文的 `font-size` 是 14 像素，`line-height` 是 20 像素。所有段落的 `margin-bottom` 是 10 像素，也就是 `line-height` 的一半。

## 2.1.1　标题

Bootstrap 为所有 6 个级别的标题都定义了样式（见图 2-1），其中 `<h1>` 是 36 像素高，`<h6>` 是 12 像素高（前面说过，正文是 14 像素高）。此外，如果想给这些标题添加行内级的子标题，可以使用 `<small>` 标签，结果文本会稍小一些，颜色会稍浅一些。拿 `<h1>` 来说，它的子标题的文本是 24 像素高，不加粗，而且是灰色：

```
h1 small {
        font-size:24px;
        font-weight:normal;
        line-height:1;
        color:#999;
}
```




图 2-1：标题

## 2.1.2　强调段落

要想强调某个段落的文本，可以为它加上 class="lead"（见图 2-2）。这会导致段落文本稍稍变大、变细，行高也变高。这种样式一般用于文章的前几段，但用在其他位置也未尝不可：

```
<p class="lead">Bacon ipsum dolor sit amet tri-tip pork loin ball tip frankfurter
swine boudin meatloaf shoulder short ribs cow drumstick beef jowl.
Meatball chicken sausage tail, kielbasa strip steak turducken venison prosciutto.
Chuck filet mignon tri-tip ribeye, flank brisket leberkas. Swine turducken
turkey shank, hamburger beef ribs bresaola pastrami venison rump.</p>
```




图 2-2：强调正文用 .lead 类

## 2.1.3　强调内容

前面讨论过可以在标题中使用 <small> 标签，在正文中也可以使用该标签。为正文中的文本添加这个标签后，文本会缩小到原来大小的 85%。

## 2.1.4　粗体

要强调具体的文本，可以使用 <strong> 标签，这样就会给相应文本应用 font-weight:bold;。

## 2.1.5　斜体

要生成斜体，使用 <em> 标签。这个标签的意思也是在文本中强调某些内容。

有读者可能不解，为什么不使用 <b> 或 <i>，而非要用 <strong> 和 <em> 呢？在 HTML5 中，<b> 的语义是仅仅突出显示某些词或短语，但不代表它们更重要，比如一些关键词或人名。而 <i> 主要用于表示不同的语言、技术术语、内部对话等。要了解 <b> 和 <i> 的语义变化，请参考 W3.org 的文章（http://www.w3.org/International/questions/qa-b-and-i-tags）。

## 2.1.6　强调相关的类

除了 <strong> 和 <em>，Bootstrap 还提供了一些表示强调的类（见图 2-3）。这些类可以应用给 <p> 或 <span>：

```
<p class="muted">This content is muted</p>
<p class="text-warning">This content carries a warning class</p>
<p class="text-error">This content carries an error class</p>
<p class="text-info">This content carries an info class</p>
<p class="text-success">This content carries a success class</p>
<p>This content has <em>emphasis</em>, and can be <strong>bold</strong></p>
```

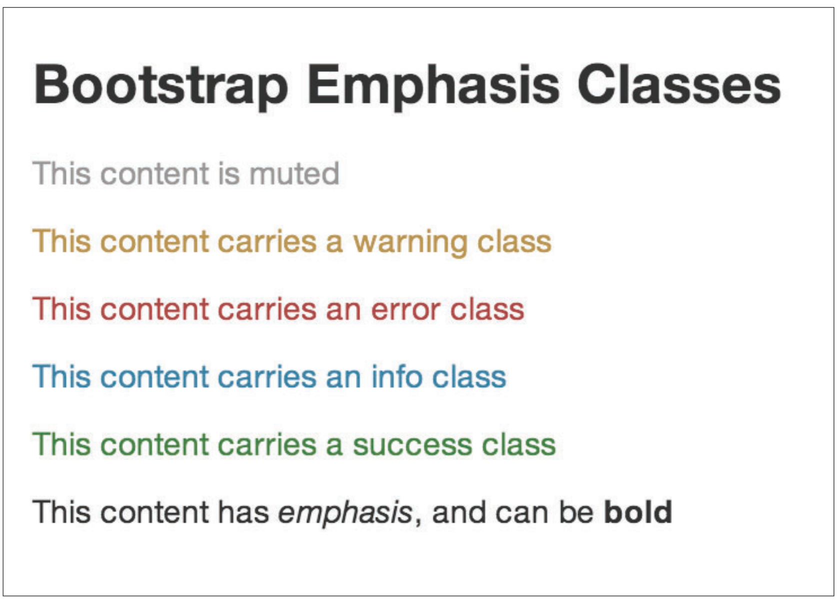



图 2-3：强调相关的类

### 1. 缩写词

HTML 的 <abbr> 元素用于标记缩写词或首字母缩写，如 WWW、HTTP（见图 2-4）。标记了缩写词之后，浏览器、拼写检查器、翻译系统或搜索引擎都能据以得知文本的性质。Bootstrap 会在 <abbr> 元素下方添加一条浅色的虚线，并在光标悬停时显示完整文本的内容（当然，前提是你在 <abbr> 的 title 属性里添加了完整的内容）：

```
<abbr title="Real Simple Syndication">RSS</abbr>
```

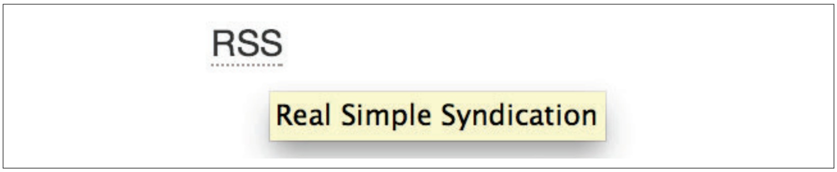



图 2-4：缩写词

为 <abbr> 添加 .initialism 类可以得到稍小一些的文本（见图 2-5）：

```
That joke had me <abbr title="rolling on the floor, laughing out loud"> ROTFLOL
</abbr>
```




图 2-5：稍小一些的缩写

### 2. 地址

在页面中使用 <address> 元素可以让屏幕阅读器和搜索引擎找到地址或电话号码（见图 2-6）。这个标签也可以标记电子邮件地址。<address> 默认是 display:block;，因此必要时可以用 <br> 标签来换行（例如把城市与街道分开）：

```
<address>
  <strong>O'Reilly Media, Inc.</strong><br>
```

```
    1005 Gravenstein HWY North<br>
    Sebastopol, CA 95472<br>
    <abbr title="Phone">P:</abbr>
    <a href="tel:+17078277000">(707) 827-7000</a>
  </address>

  <address>
    <strong>Jake Spurlock</strong><br>
    <a href="mailto:#">flast@oreilly.com</a>
  </address>
```

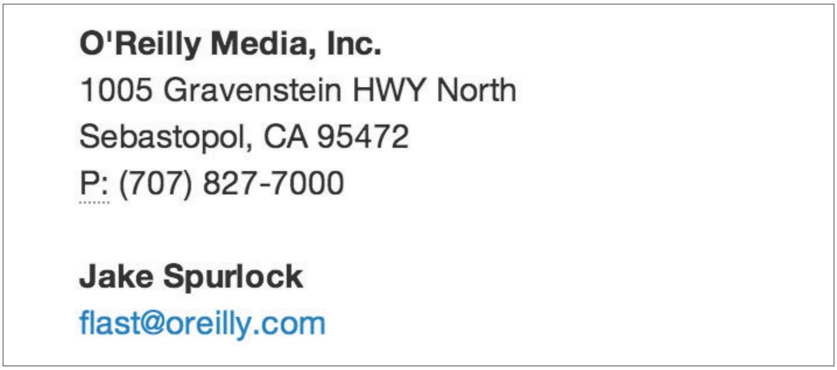


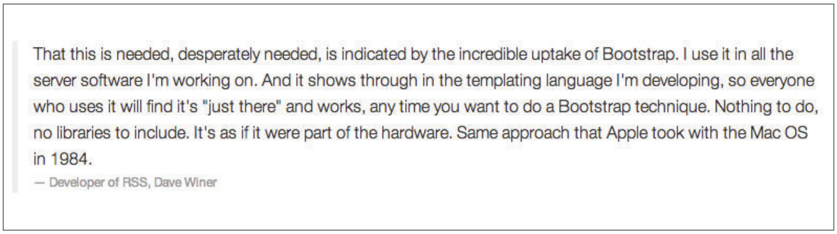

图 2-6：标记地址

### 3. 引用

要在页面中标记引用或名人名言的文本块，使用 <blockquote> 标签。为了保证效果，必要时可以添加 <br> 换行，用 <p> 分段。Bootstrap 默认会缩进引用的文本，并在左侧加一条粗边框。标记引用来源可以用 <small> 标签，人名可以用 <cite> 标签：

```
  <blockquote>
          <p>That this is needed, desperately needed, is indicated by the
      incredible uptake of Bootstrap. I use it in all the server software
      I'm working on. And it shows through in the templating language I'm
      developing, so everyone who uses it will find it's "just there" and
      works, any time you want to do a Bootstrap technique. Nothing to do,
      no libraries to include. It's as if it were part of the hardware.
      Same approach that Apple took with the Mac OS in 1984.</p>
          <small>Developer of RSS, <cite title="Source Title">Dave Winer</cite>
      </small>
  </blockquote>
```

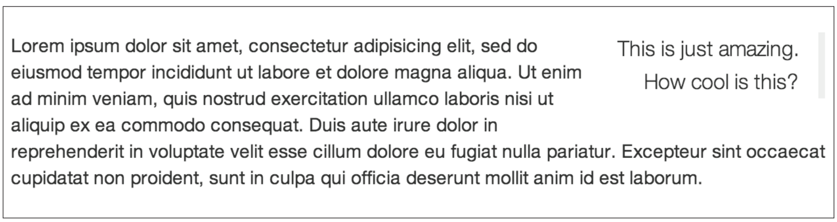
图 2-7 展示了用到所有这些标签的例子：


That this is needed, desperately needed, is indicated by the incredible uptake of Bootstrap. I use it in all the server software I'm working on. And it shows through in the templating language I'm developing, so everyone who uses it will find it's "just there" and works, any time you want to do a Bootstrap technique. Nothing to do, no libraries to include. It's as if it were part of the hardware. Same approach that Apple took with the Mac OS in 1984.

— Developer of RSS, Dave Winer

图 2-7：基本的引用

> 要想让引用右对齐，可以给 `<blockquote>` 添加 `.pull-right` 类。添加这个类后，不光是文本会右对齐，整个引用块都会浮动到右边。这样就能得到很漂亮的突出引用效果，如图 2-8 所示。


This is just amazing.

How cool is this?

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

图 2-8：突出引用

## 2.1.7　列表

Bootstrap 支持 HTML 提供的三种列表：有序列表、无序列表和定义列表。无序列表项没有特定顺序，一般样式是添加圆点符号。

### 1. 无序列表

如果想去掉无序列表前面的圆点，可以在相应的 `<ul>` 标签中添加 `class="unstyled"`（见图 2-9）：

```
<h3>Favorite Outdoor Activities</h3>
<ul>
```

```
		<li>Backpacking in Yosemite</li>
		<li>Hiking in Arches
			<ul>
				<li>Delicate Arch</li>
				<li>Park Avenue</li>
			</ul>
		</li>
		<li>Biking the Flintstones Trail</li>
</ul>
```




图 2-9：无序列表

我个人非常讨厌使用 <br> 标签。因此，遇到需要换行的情况，我都是使用没有样式的无序列表代替。比如，像图 2-6 所示的地址块，如果你想让行与行之间靠得更近些，可以把每一行都放到一个 <li> 标签中。我觉得这样能更有效地传达文本的语义。

### 2. 有序列表

有序列表项是有某种次序关系的，因此样式上使用数字而不是圆点（见图 2-10）。有序列表适合任务清单、操作步骤，甚至博客的评论列表：

```
<h3>Self-Referential Task List</h3>
<ol>
	<li>Turn off the internet.</li>
	<li>Write the book.</li>
	<li>... Profit?</li>
</ol>
```

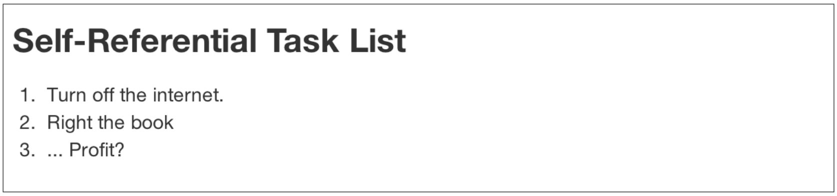



图 2-10：有序列表

### 3．定义列表

第三种列表是定义列表。定义列表与无序列表和有序列表的区别在于，它没有块级的 <li> 元素，而是每个列表项都由 <dt> 和 <dd> 元素共同组成。<dt> 代表“definition term”（词条），类似词典里收录的词语（或短语），<dd> 自然就是 <dt> 的定义。

我们经常看到有人在无序列表中加标题，虽然也可以，但从语义上讲这不是标记文本的最佳方式。更好的方法就是使用 <dl>，然后把原来应用给标题和文本的样式应用给 <dt> 和 <dd> 元素（见图 2-11）。Bootstrap 为词条和解释定义了清晰的样式，不仅可以像这样分行显示，还可以并排显示：

```
<h3>Common Electronics Parts</h3>
<dl>
        <dt>LED</dt>
        <dd>A light-emitting diode (LED) is a semiconductor light source.</dd>
        <dt>Servo</dt>
        <dd>Servos are small, cheap, mass-produced actuators used for radio
    control and small robotics.</dd>
</dl>
```

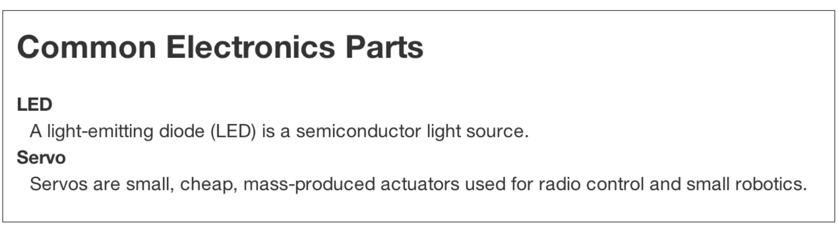



图 2-11：定义列表

要让词条和解释并排显示，只要给 `<dl>` 标签加上 `class="dl-horizontal"` 即可（见图 2-12）。

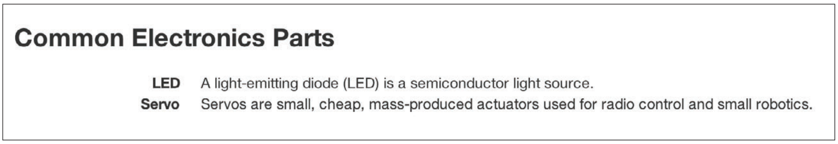



图 2-12：水平布局的定义列表

定义列表在水平布局时，词条如果过长而左边放不下，将被 `text-overflow` 属性截短。此外，在视口较窄的时候，词条和解释将自动变成默认垂直堆叠布局。

## 2.2　代码

在 Bootstrap 中有两种方式显示代码，一种是使用 `<code>` 标签，另一种是使用 `<pre>` 标签。一般来说，行内代码使用前者，代码段使用后者：

```
<p>Instead of always using divs, in HTML5, you can use new elements like
<code>&lt;section&gt;</code>, <code>&lt;header&gt;</code>, and
<code>&lt;footer&gt;</code>. The html should look something like this:</p>
<pre>
  &lt;article&gt;
    &lt;h1&gt;Article Heading&lt;/h1&gt;
  &lt;/article&gt;
</pre>
```

使用 `<pre>` 和 `<code>` 标签标记代码时，一定要把代码中的左尖括号和右尖括号替换成 `<` 和 `>`。

## 2.3　表格

Bootstrap 对表格的处理是我感觉最到位的。表格经常花掉我很多时间，而 Bootstrap 让我们轻易就能做出清爽漂亮的表格。表 2-1 列出了

Bootstrap 支持的表格元素。

表2-1　Bootstrap支持的表格元素

| 标　　签 | 作　　用 |
| --- | --- |
| <table> | 包装元素，用于显示表列式数据 |
| <thead> | 表头行（<tr>）的容器 |
| <tbody> | 表格行（<tr>）的容器 |
| <tr> | 单元格（<td> 或 <th>）容器 |
| <td> | 普通单元格 |
| <th> | 一行或一列（取决于作用范围和位置）中作为表头的单元格 |
| <caption> | 用于包装表格的说明或小结，对屏幕阅读器特别有用 |

如果你想要一个简单的表格，内容间有少许内边距，同时还有水平的分隔线，只要给 <table> 添加 .table 类即可（见图 2-13）。这种基本布局给每个 <td> 上方添加了边框：

```
<table class="table">
  <caption>...</caption>
  <thead>
    <tr>
      <th>...</th>
      <th>...</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>...</td>
      <td>...</td>
    </tr>
  </tbody>
</table>
```

| Name | Phone Number | Rank |
| --- | --- | --- |
| Kyle West | 707-827-7001 | Eagle |
| Davey Preston | 707-827-7003 | Eagle |
| Taylor Lemmon | 707-827-7005 | Eagle |

图 2-13：基本的 .table 类

## 2.3.1　可选的表格类

除了基本的表格标记和 `.table` 类，Bootstrap 还提供了 4 个类为表格添加不同的样式，它们是：`.table-striped`、`.table-bordered`、`.table-hover` 和 `.table-condensed`。

### 1．带条纹背景的表格

添加 `.table-striped` 类，就可以为 `<tbody>` 中的表格行加上条纹背景（见图 2-14）。这是通过 `:nth-child` 选择符实现的，但 Internet Explorer 7 和 8 不支持这个选择符。

| Name | Phone Number | Rank |
|---|---|---|
| Kyle West | 707-827-7001 | Eagle |
| Davey Preston | 707-827-7003 | Eagle |
| Taylor Lemmon | 707-827-7005 | Eagle |

图 2-14：带条纹背景的表格

### 2．带边框的表格

添加 `.table-bordered` 类，可以为所有单元格加上边框，为整个表格加上圆角，如图 2-15 所示。

| Name | Phone Number | Rank |
|---|---|---|
| Kyle West | 707-827-7001 | Eagle |
| Davey Preston | 707-827-7003 | Eagle |
| Taylor Lemmon | 707-827-7005 | Eagle |

图 2-15：带边框的表格

### 3．带悬停高亮的表格

图 2-16 展示了添加 `.table-hover` 类的效果，此时光标悬停的行会带有浅灰色背景。

| Name | Phone Number | Rank |
| --- | --- | --- |
| Kyle West | 707-827-7001 | Eagle |
| Davey Preston | 707-827-7003 | Eagle |
| Taylor Lemmon | 707-827-7005 | Eagle |

图 2-16：带悬停高亮的表格

### 4．紧缩的表格

添加 `.table-condensed` 类，如图 2-17 所示，就会减少一半的内边距，以紧缩表格。这种样式适合信息密集的情况。

| Name | Phone Number | Rank |
| --- | --- | --- |
| Kyle West | 707-827-7001 | Eagle |
| Davey Preston | 707-827-7003 | Eagle |
| Taylor Lemmon | 707-827-7005 | Eagle |

图 2-17：紧缩的表格

## 2.3.2　表格行的类

表 2-2 列出了可以添加给表格行的类，用于改变特定行的颜色（见图 2-18）。

表2-2　表格行可选的类

| 类 | 说　明 | 背景颜色 |
| --- | --- | --- |
| `.success` | 表示成功或积极的情形 | 绿 |
| `.error` | 表示危险或消极的情形 | 红 |
| `.warning` | 表示警告 | 黄 |
| `.info` | 用于代替默认样式 | 蓝 |

| # | Product | Payment Taken | Status |
| --- | --- | --- | --- |
| 1 | TB - Monthly | 01/04/2012 | Approved |
| 2 | TB - Monthly | 02/04/2012 | Declined |
| 3 | TB - Monthly | 03/04/2012 | Pending |
| 4 | TB - Monthly | 04/04/2012 | Call in to confirm |

图 2-18：可应用于表格行的类

## 2.4 表单

表单也是 Bootstrap 的重头戏，让开发人员得到了解放。作为 Web 开发人员，我最不喜欢设计表单样式了。而使用 Bootstrap，只要一些简单的 HTML 标记和一些类，就能做出不同风格的表单。

Bootstrap 自带了基本的表单样式，不需要添加任何辅助类（见图 2-19）。如果你使用占位符，请注意只有新版本浏览器才支持。旧版本浏览器不会显示占位符：

```
<form>
        <fieldset>
                <legend>Legend</legend>
                        <label for="name">Label name</label>
                        <input type="text" id="name"
                   placeholder="Type something...">
                        <span class="help-block">Example block-level help
                         text here.</span>
                        <label class="checkbox" for="checkbox">
                                <input type="checkbox" id="checkbox">
                Check me out
                        </label>
                <button type="submit" class="btn">Submit</button>
        </fieldset>
</form>
```

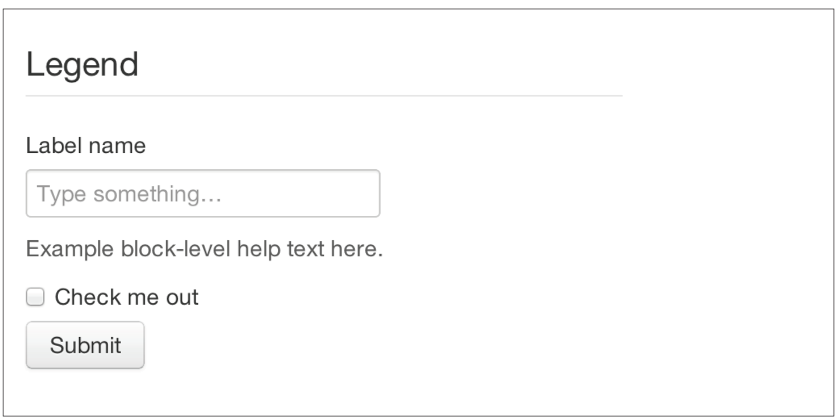



图 2-19：基本的表单

## 2.4.1 可选的表单布局

通过一些辅助类，可以动态更新表单的布局。Bootstrap 为常用的表单布局预定义了样式。

### 1．搜索表单

为 `<form>` 标签添加 `.form-search` 类，再给 `<input>` 添加 `.search-query` 类，然后就可以创建一个圆角输入框加一个行内提交按钮的搜索表单（见图 2-20）：

```
<form class="form-search">
  <input type="text" class="input-medium search-query">
  <button type="submit" class="btn">Search</button>
</form>
```




图 2-20：搜索表单

### 2．行内表单

要让所有表单元素都出现在一行，而且标签也都显示在旁边，给 `<form>` 标签添加 `.form-inline` 类（见图 2-21）。下面的示例代码说明了如何创建行内表单：

```
<form class="form-inline">
        <input type="text" class="input-small" placeholder="Email">
        <input type="password" class="input-small" placeholder="Password">
        <label class="checkbox">
                <input type="checkbox"> Remember me
        </label>
        <button type="submit" class="btn">Sign in</button>
</form>
```

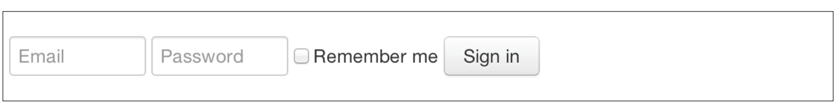



图 2-21：行内表单

### 3．横向表单

Bootstrap 还支持预定义的横向表单，这种表单涉及的标记多，外观也独树一帜。以前，人们大都使用表格来布局图 2-22 所示的这种表单，Bootstrap 抛开表格实现了这种布局。而且，如果你使用了响应式 CSS，横向表单还能自动适应较小的屏幕，把所有控件堆叠起来。

要创建横向布局的表单，请遵循如下步骤：

- 为 `<form>` 元素添加 `.form-horizontal` 类；
- 把成对的标签和控件包装在类为 `.control-group` 的 `<div>` 中；
- 为其中每个标签添加 `.control-label` 类；
- 把其中每个控件包装在一个类为 `.controls` 的 `<div>` 中，以便对齐。




图 2-22：横向表单

```
<form class="form-horizontal">
  <div class="control-group">
    <label class="control-label" for="inputEmail">Email</label>
    <div class="controls">
      <input type="text" id="inputEmail" placeholder="Email">
    </div>
  </div>
  <div class="control-group">
    <label class="control-label" for="inputPassword">Password</label>
    <div class="controls">
      <input type="password" id="inputPassword" placeholder="Password">
    </div>
  </div>
  <div class="control-group">
    <div class="controls">
      <label class="checkbox">
```

```
      <input type="checkbox"> Remember me
    </label>
    <button type="submit" class="btn">Sign in</button>
  </div>
</div>
</form>
```

## 2.4.2　内置支持的表单控件

Bootstrap 原生支持各种最常用的表单控件，主要包括输入框、文本区、复选框、单选按钮和选项列表。

### 1. 输入框

最常用的文本字段就是输入框，用户要通过它们输入大多数表单数据（见图 2-23）。Bootstrap 支持 HTML5 规范定义的所有原生输入类型：`text`、`password`、`datetime`、`datetime-local`、`date`、`month`、`time`、`week`、`number`、`email`、`url`、`search`、`tel` 和 `color`。比如：

```
<input type="text" placeholder="Text input">
```

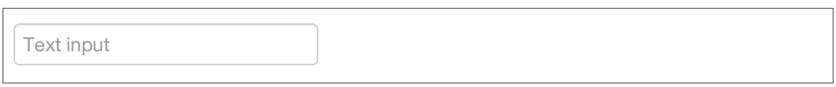
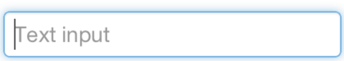



图 2-23：输入框

输入框和文本区在 `:active` 状态默认都带有蓝色光晕。




### 2. 文本区

文本区用于输入多行文本（见图 2-24）。使用文本区时，最经常做的就是修改其 `rows`，控制预留行数（行数少，占地方小；行数多，占地方大）：

```
<textarea rows="3"></textarea>
```

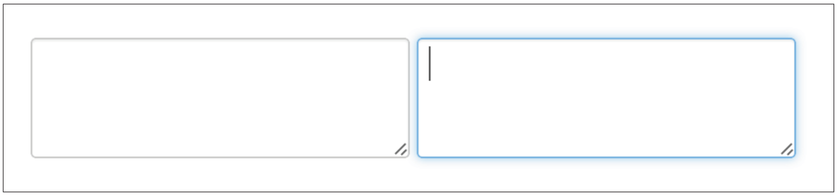


图 2-24：默认状态和 :active 状态的文本区

### 3. 复选框和单选按钮

复选框和单选按钮非常适合让用户在一组预定项中选择（见图 2-25）。如果想让用户选择任意多项，使用 <checkbox>，如果想让用户只选择一项，使用 <radio>：

```
<label class="checkbox">
  <input type="checkbox" value="">
  Option one is this and that—be sure to include why it's great.
</label>

<label class="radio">
  <input type="radio" name="optionsRadios" id="optionsRadios1" value=
  "option1" checked>
  Option one is this and that—be sure to include why it's great.
</label>
<label class="radio">
  <input type="radio" name="optionsRadios" id="optionsRadios2" value=
  "option2">
  Option two can be something else, and selecting it will deselect option one
</label>
```

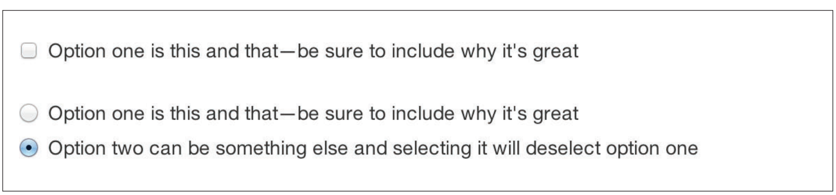



图 2-25：复选框和单选按钮

如果想让多个复选框或单选按钮出现在一行，那么就给它们添加 .inline 类（见图 2-26）：

```
<label for="option1" class="checkbox inline">
  <input id="option1" type="checkbox" id="inlineCheckbox1" value="option1"> 1
</label>
<label for="option2" class="checkbox inline">
  <input id="option2" type="checkbox" id="inlineCheckbox2" value="option2"> 2
</label>
<label for="option3" class="checkbox inline">
  <input id="option3" type="checkbox" id="inlineCheckbox3" value="option3"> 3
</label>
```




图 2-26：行内复选框

## 4．选项列表

选项列表适合让用户选择多项，但默认只允许选择一项（见图 2-27）。建议只在用户对选项很熟悉的情况下使用 <select>，如州名或数值。要让用户选择多项，使用 multiple="multiple"：

```
<select>
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>

<select multiple="multiple">
  <option>1</option>
  <option>2</option>
  <option>3</option>
  <option>4</option>
  <option>5</option>
</select>
```




图 2-27：选项列表

## 2.4.3　扩展的表单控件

除了上一节介绍的基本表单控件，Bootstrap 还提供了另外几个控件，作为对 HTML 表单元素的补充。比如，输入框的前置和后置组件。

### 输入框的前置和后置组件

利用输入框的前置和后置组件，可以添加用户输入中的公共部分（见图 2-28）。例如，可以为输入框前置美元符号、Twitter 用户名的 @ 符号，或应用界面中其他公共的部分。要在输入框前面或后面添加内容，首先写一个类为 `.input-prepend` 或 `.input-append` 的 `<div>`。然后就在这个 `<div>` 中，把要前置或后置的内容放到一个类为 `.add-on` 的 `<span>` 中，再把这个 `<span>` 放到相应的 `<input>` 元素前面或后面：

```
<div class="input-prepend">
  <span class="add-on">@</span>
  <input class="span2" id="prependedInput" type="text"
       placeholder="Username">
</div>
<div class="input-append">
  <input class="span2" id="appendedInput" type="text">
  <span class="add-on">.00</span>
</div>
```

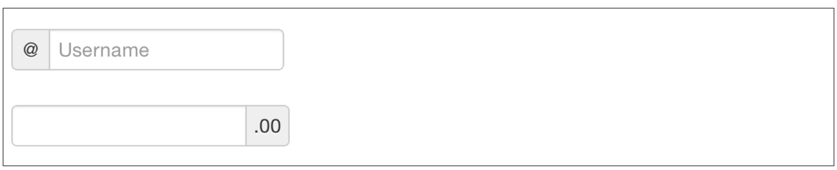



图 2-28：前置和后置内容

如果既想前置内容又想后置内容，那就在父 `<div>` 中同时添加 `.input-prepend` 和 `.input-append` 类（见图 2-29）：

```
<div class="input-prepend input-append">
  <span class="add-on">$</span>
  <input class="span2" id="appendedPrependedInput" type="text">
  <span class="add-on">.00</span>
</div>
```

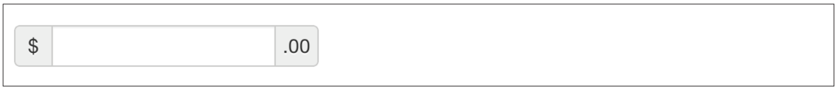



图 2-29：同时前置和后置内容

此外，还可以使用带 .btn 类的 <button> 代替 <span>，在输入框前面和 / 或后面添加按钮（见图 2-30）：

```
<div class="input-append">
  <input class="span2" id="appendedInputButtons" type="text">
  <button class="btn" type="button">Search</button>
  <button class="btn" type="button">Options</button>
</div>
```




图 2-30：在输入框后面添加多个按钮

要是给搜索表单后置这么一个按钮，那么按钮也会像文本框一样带有圆弧（见图 2-31）：

```
<form class="form-search">
  <div class="input-append">
    <input type="text" class="span2 search-query">
    <button type="submit" class="btn">Search</button>
  </div>
  <div class="input-prepend">
    <button type="submit" class="btn">Search</button>
    <input type="text" class="span2 search-query">
  </div>
</form>
```

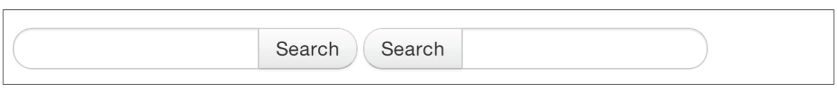



图 2-31：为搜索表单添加按钮

### 2.4.4　设置控件大小

在继承 Bootstrap 默认网格系统的基础上，可以使用 .span* 来设置

控件大小。除了这种方式，还可以使用一些类来相对地控制控件大小。如果想让输入框像块级元素一样占满容器，可以为它添加 `.input-block-level` 类，如图 2-32 所示：

```
<input class="input-block-level" type="text" placeholder=".input-block-level">
```

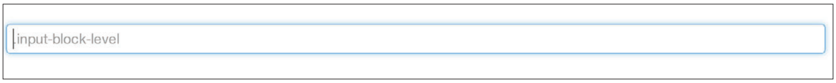



图 2-32：块级输入框

### 1．设置输入控件的相对大小

如前所述，设置控件大小时除了使用 `.span*`，还可以使用一些类名（见图 2-33）：

```
<input class="input-mini" type="text" placeholder=".input-mini">
<input class="input-small" type="text" placeholder=".input-small">
<input class="input-medium" type="text" placeholder=".input-medium">
<input class="input-large" type="text" placeholder=".input-large">
<input class="input-xlarge" type="text" placeholder=".input-xlarge">
<input class="input-xxlarge" type="text" placeholder=".input-xxlarge">
```




图 2-33：相对大小的输入控件

在 Bootstrap 将来的版本中，这些类的样式还会加以修正，以匹配按钮大小。比如，`.input-large` 会同时增大输入框的内边距和文本大小。

### 2．基于网格设置控件大小

还可以使用 .span1 到 .span12 来基于网格设置控件大小（见图 2-34）：

```
<input class="span1" type="text" placeholder=".span1">
<input class="span2" type="text" placeholder=".span2">
<input class="span3" type="text" placeholder=".span3">
<select class="span1">
  ...
</select>
<select class="span2">
  ...
</select>
<select class="span3">
  ...
</select>
```




图 2-34：跨列的输入控件

如果想让多个输入控件位列同一行，可以添加 .controls-row 类，以产生适当间距（见图 2-35）。这样 Bootstrap 就会浮动控件，设置间距，并像添加 .row 类一样清除浮动：

```
<div class="controls">
  <input class="span5" type="text" placeholder=".span5">
</div>
<div class="controls controls-row">
  <input class="span4" type="text" placeholder=".span4">
  <input class="span1" type="text" placeholder=".span1">
</div>
...
```




图 2-35：一行控件

### 3．不可编辑的文本

如果想展示一个不让用户编辑的表单控件，只要添加 .uneditable-input 类即可（见图 2-36）：

```
<span class="input-xlarge uneditable-input">Some value here</span>
```




图 2-36：不可编辑的输入控件

### 4．表单动作

如果在 .form-horizontal 最后放上几个 .form-actions（也就是按钮），那么这些按钮会自动与表单控件对齐（见图 2-37）：

```
<div class="form-actions">
  <button type="submit" class="btn btn-primary">Save changes</button>
  <button type="button" class="btn">Cancel</button>
</div>
```




图 2-37：表单控件

### 5．帮助文本

Bootstrap 中的表单控件可以带有块级或行内文本作为帮助（见图 2-38）：

```
<input type="text"><span class="help-inline">Inline help text</span>
```




图 2-38：行内帮助

要添加块级帮助，为 <input> 后面的标签添加 .help-block 类（见图 2-39）：

```
<input type="text"><span class="help-block">A longer block of help text
that breaks onto a new line and may extend beyond one line.</span>
```




图 2-39：块级帮助

## 2.4.5　表单控件的状态

除了 :focus 状态的样式外，Bootstrap 还为被禁用的输入控件提供了样式，为表单验证提供了一些类。

### 1．输入控件获得焦点

输入控件获得焦点时（比如用户单击了输入框或按 Tab 键切换到输入框），输入框默认的外轮廓样式消失，代之以 box-shadow 样式。我还记得第一次在 Twitter 上看到这个效果时的情景，我想不出来是怎么实现的，禁不住去查看了它的源代码。在 WebKit 中，这个效果可以这样实现：

```
input {
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075);
  -webkit-transition: box-shadow linear 0.2s;
}

input:focus {
  -webkit-box-shadow: inset 0 1px 1px rgba(0, 0, 0, 0.075), 0 0 8px
                      rgba(82, 168, 236, 0.6);
}
```

这里给 `<input>` 添加了少量内阴影，让人觉得输入框有点下凹（见图 2-40）。而在 `:focus` 状态下，则会应用 8 像素的浅蓝色边框。`webkit-transition` 告诉浏览器以 0.2 秒的时间匀速完成动画：

```
<input class="input-xlarge" id="focusedInput" type="text"
              value="This is focused...">
```




图 2-40：获得焦点的输入框

精细微妙，多好的效果啊！

## 2．禁用状态的输入框

要禁用某个输入框，只要给 `<input>` 加上 `disabled` 属性即可，这样也会改变悬停光标的样式（见图 2-41）：

```
<input class="input-xlarge" id="disabledInput" type="text"
              placeholder="Disabled input here..." disabled>
```




图 2-41：禁用状态的输入框

## 3．验证状态

Bootstrap 提供的验证样式包括 `error`（错误）、`warning`（警告）、`info`

（信息）和 success（成功），如图 2-42 所示。要应用这些样式，只要把相应的类添加给 .control-group 元素即可：

```
<div class="control-group warning">
  <label class="control-label" for="inputWarning">Input with warning</label>
  <div class="controls">
    <input type="text" id="inputWarning">
    <span class="help-inline">Something may have gone wrong</span>
  </div>
</div>
<div class="control-group error">
  <label class="control-label" for="inputError">Input with error</label>
  <div class="controls">
    <input type="text" id="inputError">
    <span class="help-inline">Please correct the error</span>
  </div>
</div>
<div class="control-group success">
  <label class="control-label" for="inputSuccess">Input with success</label>
  <div class="controls">
    <input type="text" id="inputSuccess">
    <span class="help-inline">Woohoo!</span>
  </div>
</div>
```

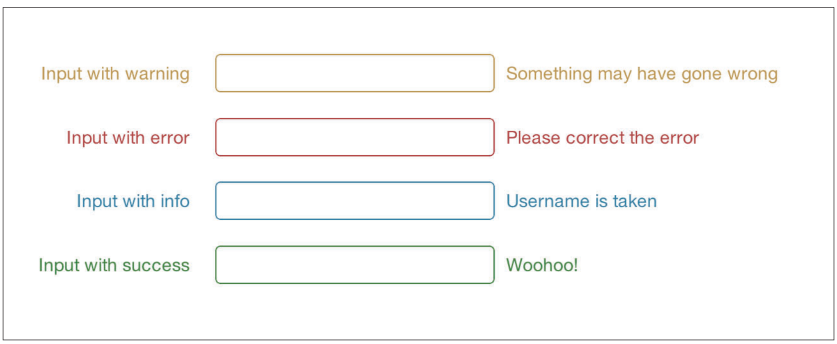



图 2-42：验证状态

## 2.5 按钮

Bootstrap 为按钮提供了很多漂亮的样式。RSS 之父 Dave Winer 也是 Bootstrap 的忠实粉丝，他曾说过：

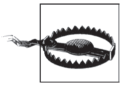
> Bootstrap是必需的、绝对必要的，它的迅速普及就是明证。不光我的服务器端软件会用到它，我开发的模板语言里也少不了它，这样只要用户想使用Bootstrap，随时随地用就可以了。什么也不用做，也不用包含什么库。就像服务器硬件的一部分，跟苹果1984年在Mac OS上所做的一样。
>
> ——Dave Winer
> scripting.com

我认为 Bootstrap 最大的好处是统一 Web 界面和体验。有了 Bootstrap，你在任何站点上一眼就能看出它使用了什么按钮。网格系统和其他很多功能都隐藏在后台，但按钮、表单，以及其他外观一致的元素也是 Bootstrap 的重要组成部分。每当看到使用 Bootstrap 的网站，我都有一种跟站长击掌致意的冲动，就像几年前我在某些网站源代码中看到 wp-content 时一样。

现在，按钮和链接在 Bootstrap 中的外观很像。任何带 `.btn` 类的元素都会继承灰色圆角按钮的样式。不过，使用其他类是可以给按钮加上颜色的（见表 2-3）。

表2-3　彩色按钮示例

| 按　钮 | `class="..."` | 说　　明 |
| --- | --- | --- |
| Default | `btn` | 带渐变的灰色标准按钮 |
| Primary | `btn btn-primary` | 表示按钮的操作相对比较重要 |
| Info | `btn btn-info` | 用于代替默认的标准按钮 |
| Success | `btn btn-success` | 表示成功或积极的操作 |
| Warning | `btn btn-warning` | 表示对相应操作要格外小心 |
| Danger | `btn btn-danger` | 表示危险或消极的操作 |
| Inverse | `btn btn-inverse` | 用于代替深灰色按钮，无关语义 |
| Link | `btn btn-link` | 让按钮看起来像链接，同时还具备按钮的行为 |



由于 IE9 不能剪切掉圆角处的背景渐变，因此 IE9 中删除了渐变效果。IE 在显示被禁用的按钮时也有问题，比如文本是灰色，而且还带有难看的文本阴影。

### 2.5.1　按钮大小

如果你希望按钮大一些或小一些，可以给链接或按钮添加 `.btn-large`、`.btn-small` 或 `.btn-mini` 类（见图 2-43）：

```
<p>
  <button class="btn btn-large btn-primary" type="button">Large button
  </button>
  <button class="btn btn-large" type="button">Large button</button>
</p>
<p>
  <button class="btn btn-primary" type="button">Default button</button>
  <button class="btn" type="button">Default button</button>
</p>
<p>
  <button class="btn btn-small btn-primary" type="button">Small button
  </button>
  <button class="btn btn-small" type="button">Small button</button>
</p>
<p>
  <button class="btn btn-mini btn-primary" type="button">Mini button
  </button>
  <button class="btn btn-mini" type="button">Mini button</button>
</p>
```

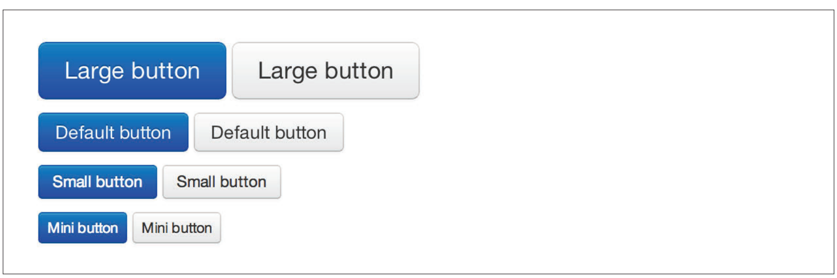



图 2-43：不同大小的按钮

如果想创建一个“块级”按钮，也没问题，添加 .btn-block 类（见图 2-44）。此时的按钮宽度是 100%：

```
<button class="btn btn-large btn-block btn-primary" type="button">
    Block-level button</button>
<button class="btn btn-large btn-block" type="button">Block-level button
</button>
```

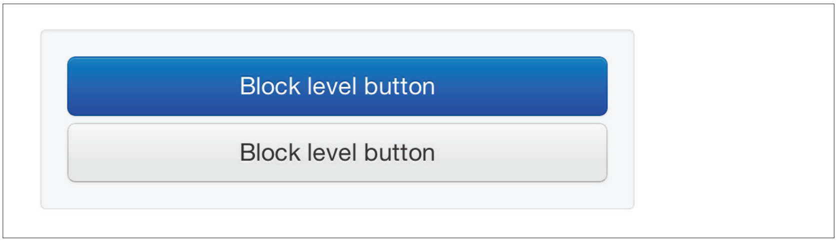



图 2-44：块级按钮

## 2.5.2　禁用按钮的样式

对于链接，只要添加 .disabled 类就可以实现颜色变浅、褪掉渐变的效果（见图 2-45）：

```
<a href="#" class="btn btn-large btn-primary disabled">Primary link</a>
<a href="#" class="btn btn-large disabled">Link</a>
```

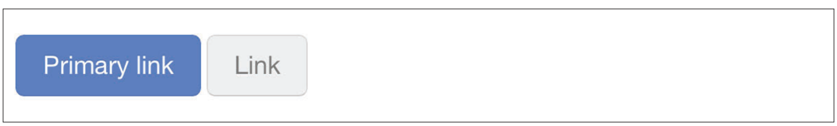



图 2-45：禁用的链接

注意，.disabled 类的用法与 .active 类很相似。也就是说，不需要 .btn 前缀，但要记住这只会影响样式。如果你真要禁用链接，还得让 JavaScript 帮忙。

对于按钮，添加 disabled 属性就行了（见图 2-46）。这样就会禁用按钮，用不着 JavaScript：

```
<button type="button" class="btn btn-large btn-primary disabled"
disabled="disabled">Primary button</button>
<button type="button" class="btn btn-large" disabled>Button</button>
```




图 2-46：禁用的按钮

## 2.6 图片

与图片相关的类有 3 个（见图 2-47），可用于为图片应用一些简单的样式：.img-rounded 为图片的四角添加 border-radius:6px、.img-circle 通过添加 border-radius:500px 使图片变成圆形、.img-polaroid 会添加一点内边距和一条灰色边框：

```
<img src="..." class="img-rounded">
<img src="..." class="img-circle">
<img src="..." class="img-polaroid">
```

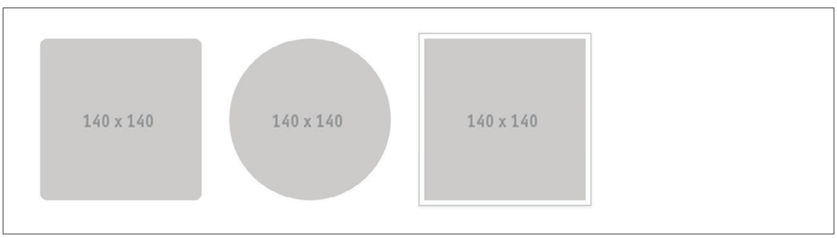



图 2-47：图片

## 2.7 图标

Bootstrap 附带了一张 140 个图标的精灵图（sprite），这些图标可用于按钮、链接、导航和表单字段。这些图标由 GLYPHICONS 提供，参见图 2-48。




图 2-48：GLYPHICONS 提供的图标

## 2.7.1　感谢GLYPHICONS提供图标

作为 Bootstrap 的用户，我们很幸运能免费使用 GLYPHICONS（http://glyphicons.com/）提供的图标。请各位开发人员在适当的情况下，声明图标来源于 GLYPHICONS。

> GLYPHICONS Halflings正常情况下是要收费的，但经过Bootstrap和GLYPHICONS 协商，广大开发人员可以免费使用这些图标。为表谢意，希望各位在合适的情况下注明图标源于GLYPHICONS并给出链接。
>
> ——Bootstrap官方文档

### 2.7.2　用法

要使用图标，给 <i> 标签添加以 .icon- 为前缀的类即可。例如，要使用编辑图标，给 <i> 标签添加 .icon-edit 类：

```
<i class="icon-edit"></i>
```

如果想使用白色图标，再补充一个 .icon-white 类即可：

```
<i class="icon-edit icon-white"></i>
```

#### 1．按钮组

同时使用按钮组和图标，可以用最少的标记设计出漂亮的界面（见图 2-49）：

```
<div class="btn-toolbar">
  <div class="btn-group">
    <a class="btn" href="#"><i class="icon-align-left"></i></a>
    <a class="btn" href="#"><i class="icon-align-center"></i></a>
    <a class="btn" href="#"><i class="icon-align-right"></i></a>
    <a class="btn" href="#"><i class="icon-align-justify"></i></a>
  </div>
</div>
```

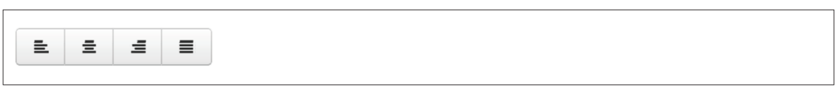


图 2-49：按钮组

#### 2．导航

在把图标放到文本旁边时，注意在图标和文本间留出适当空白（见图 2-50）。下一章还会进一步讨论关于导航的代码。

```
<ul class="nav nav-list">
  <li class="active"><a href="#"><i class="icon-home icon-white"></i>
  Home</a></li>
  <li><a href="#"><i class="icon-book"></i> Library</a></li>
  <li><a href="#"><i class="icon-pencil"></i> Applications</a></li>
  <li><a href="#"><i class="i"></i> Misc</a></li>
</ul>
```




图 2-50：基本的导航列表

## 第 3 章

# Bootstrap内置的布局组件

除了前一章介绍的标记，Bootstrap 还内置了一套灵活的组件，可用于设计用户界面、交互功能，等等。这些组件来自一个独立的 JavaScript 文件，而这个文件可以通过 Bootstrap 的自定义构建工具生成，因此可以只包含你需要的组件。我个人一般会选择所有组件，这样就有了最大的灵活性。

## 3.1　下拉菜单

下拉菜单以列表形式呈现一组选项，一般与上下文相关，可用于不同的元素、导航、按钮，等等。除了图 3-1 所示的基本菜单之外，下拉菜单还可以包含子菜单。




图 3-1：基本的下拉菜单

以下是创建这个下拉菜单的代码：

```
<ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
  <li><a tabindex="-1" href="#">Action</a></li>
  <li><a tabindex="-1" href="#">Another action</a></li>
  <li><a tabindex="-1" href="#">Something else here</a></li>
  <li class="divider"></li>
  <li><a tabindex="-1" href="#">Separated link</a></li>
</ul>
```

### 选项

#### 1．右对齐

为 `.dropdown-menu` 添加 `.pull-right` 类，可以让下拉菜单右边缘与父对象右边缘对齐：

```
<ul class="dropdown-menu pull-right" role="menu" aria-labelledby="dLabel">
  ...
</ul>
```

### 2. 子菜单

要添加像图 3-2 所示那样的子菜单，只要在当前下拉菜单的任意 `<li>` 元素上添加 `.dropdown-submenu` 即可：

```
<ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">
  ...
  <li class="dropdown-submenu">
    <a tabindex="-1" href="#">More options</a>
    <ul class="dropdown-menu">
      ...
    </ul>
  </li>
</ul>
```




图 3-2：下拉菜单与子菜单

## 3.2 按钮组

按钮组可以把多个按钮组织到一起（见图 3-3）。利用按钮组可以把几个相关的按钮（比如文本对齐按钮）放在一起。要创建按钮组，只要把相应的链接或按钮放在一个类为 `.btn-group` 的 `<div>` 中即可：

```
<div class="btn-group">
  <button class="btn">Left</button>
  <button class="btn">Middle</button>
  <button class="btn">Right</button>
</div>
```




图 3-3：包含左、中、右对齐按钮的按钮组

如果想把多个按钮组放到一行上（见图 3-4），可以用类为 .btn-toolbar 的 <div> 来包装这些按钮组：

```
<div class="btn-toolbar">
  <div class="btn-group">
    <a class="btn" href="#"><i class="icon-align-left"></i></a>
    <a class="btn" href="#"><i class="icon-align-center"></i></a>
    <a class="btn" href="#"><i class="icon-align-right"></i></a>
    <a class="btn" href="#"><i class="icon-align-justify"></i></a>
  </div>
  <div class="btn-group">
    <a class="btn" href="#"><i class="icon-italic"></i></a>
    <a class="btn" href="#"><i class="icon-bold"></i></a>
    <a class="btn" href="#"><i class="icon-font"></i></a>
    <a class="btn" href="#"><i class="icon-text-height"></i></a>
    <a class="btn" href="#"><i class="icon-text-width"></i></a>
  </div>
  <div class="btn-group">
    <a class="btn" href="#"><i class="icon-indent-left"></i></a>
    <a class="btn" href="#"><i class="icon-indent-right"></i></a>
  </div>
</div>
```

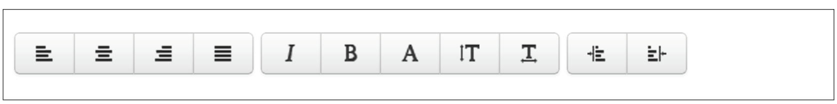


图 3-4：按钮工具栏

要了解在按钮中使用图标的更多信息，请参考第 2 章。要想把按钮垂直堆叠起来（见图 3-5），可以在 .btn-group 后面再添加一个 .btn-group-vertical 类：

```
<div class="btn-group btn-group-vertical">
  ...
</div>
```

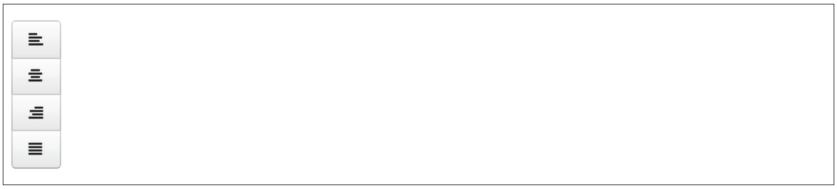


图 3-5：垂直按钮组

## 让按钮组具有单选按钮和复选框的功能

要想让按钮组具有单选按钮或复选框的功能（即每次只能选一个按钮，或每次可选择多个按钮），只要多写几行标记即可，其他的事儿 JavaScript 都会替你做好。详细信息将在第 4 章讨论。

要在下拉菜单中使用按钮，必须把按钮放在 .btn-toolbar 中独立的 .btn-group 中。

# 3.3 按钮下拉菜单

要为按钮添加下拉菜单（见图 3-6），只需把按钮和下拉菜单包装在一个 .btn-group 中即可。再加上一个 <span class="caret"></span>，就可以从视觉上告诉人们按钮包含下拉菜单：

```
<div class="btn-group">
  <button class="btn btn-danger">Danger</button>
  <button class="btn btn-danger dropdown-toggle" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">Action</a></li>
    <li><a href="#">Another action</a></li>
    <li><a href="#">Something else here</a></li>
    <li class="divider"></li>
    <li><a href="#">Separated link</a></li>
  </ul>
</div>
```

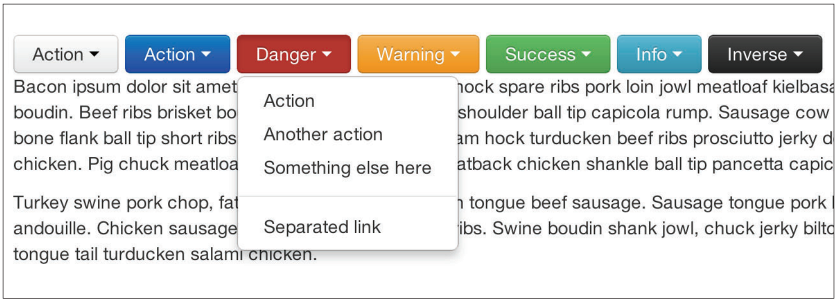



图 3-6：按钮式下拉菜单

下拉菜单可以配合任意按钮大小：`.btn-large`、`.btn`、`.btn-small` 或 `.btn-mini`。图 3-7 展示了不同大小的一些按钮。




图 3-7：不同大小的按钮与下拉菜单

## 3.3.1 分隔式按钮下拉菜单

分隔式按钮下拉菜单（见图 3-8）与按钮下拉菜单的样式相同，但有分隔线隔开左边的主操作按钮和右边的下拉菜单开关按钮。




图 3-8：分隔式按钮下拉菜单

以下是实现分隔式按钮下拉菜单的代码：

```
<div class="btn-group">
  <button class="btn">Action</button>
  <button class="btn dropdown-toggle" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <!-- dropdown menu links -->
  </ul>
</div>
```

## 3.3.2 上拉菜单

菜单不仅可以下拉，也可以上拉（见图 3-9）。要把默认的下拉改成上拉，只要在 `.btn-group` 容器中再加一个 `.dropup` 类就行了。另外，如果还想让上拉菜单与开关按钮右侧对齐，则需要在 `.dropdown-menu` 中再加一个 `.pull-right` 类（注意，开关按钮中的箭头现在冲上了，因为菜单会从上面弹出）：

```
<div class="btn-group dropup">
  <button class="btn">Dropup</button>
  <button class="btn dropdown-toggle" data-toggle="dropdown">
    <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <!-- dropdown menu links -->
  </ul>
</div>
```

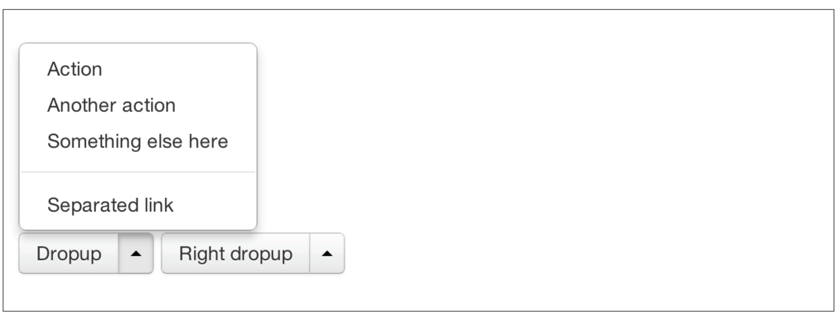



图 3-9：上拉菜单

## 3.4 导航元素

Bootstrap 提供几种不同样式的导航元素，这些元素只是样式不同，但都具有同样的标记和基准的 `.nav` 类。

Bootstrap 还提供了一个辅助的 `.active` 类。这个类一般用于区分当前元素和其他未激活的元素。可以把这个类应用给主页链接，或者用户当前所在页的链接。

### 3.4.1 标签形导航

要创建标签形导航（见图 3-10），只要给一个无序列表添加基本的 `.nav` 和 `.nav-tabs` 类即可：

```
<ul class="nav nav-tabs">
  <li class="active">
    <a href="#">Home</a>
  </li>
  <li><a href="#">Profile</a></li>
  <li><a href="#">Messages</a></li>
</ul>
```




图 3-10：标签形导航

### 3.4.2 胶囊形导航

要把标签变成胶囊的形状（见图 3-11），用 `.nav-pills` 代替 `.nav-tabs` 类即可：

```
<ul class="nav nav-pills">
  <li class="active">
    <a href="#">Home</a>
  </li>
  <li><a href="#">Profile</a></li>
  <li><a href="#">Messages</a></li>
</ul>
```

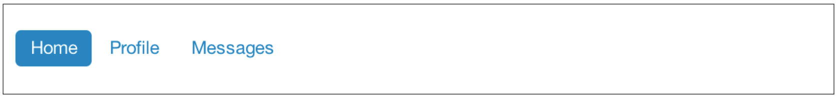



图 3-11：胶囊形导航

### 禁用导航元素

要想禁用某个链接，只要给它加上 .disabled 类，就可以让它在任何情况下（包括 :hover 状态下）都变为灰色（见图 3-12）：

```
<ul class="nav nav-pills">
  ...
  <li class="disabled"><a href="#">Home</a></li>
  ...
</ul>
```




图 3-12：禁用链接

## 3.4.3　垂直堆叠的导航

标签形和胶囊形导航元素默认都是水平排列的。要让它们垂直堆叠起来，只要再添加一个 .nav-stacked 类即可。图 3-13 和图 3-14 是垂直堆叠的标签和胶囊的例子。

以下是堆叠式导航的代码：

```
<ul class="nav nav-tabs nav-stacked">
  ...
</ul>
```




图 3-13：堆叠式标签

以下是堆叠式胶囊的代码：

```
<ul class="nav nav-pills nav-stacked">
  ...
</ul>
```

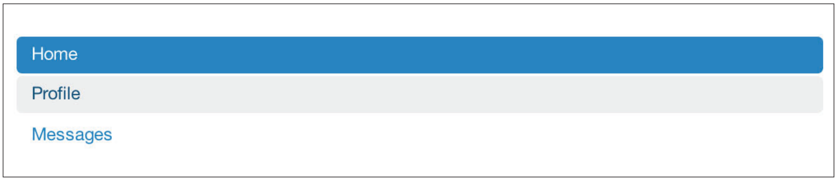



图 3-14：堆叠式胶囊

## 3.4.4　导航加下拉菜单

导航菜单与下拉菜单的语法类似（见图 3-15）。默认情况下，给一个列表项添加 .dropdown 类，然后给嵌套在这个列表项中的链接添加 .dropdown-toggle 类和 data-toggle="dropdown" 属性，再给嵌套在这个列表项中的无序列表元素添加 .dropdown-menu 类即可：

```
<ul class="nav nav-tabs">
        <li class="dropdown">
                <a class="dropdown-toggle"
                        data-toggle="dropdown"
                        href="#">
                        Dropdown
                        <b class="caret"></b>
                </a>
                <ul class="dropdown-menu">
        <li><a href="#">Action</a></li>
        <li><a href="#">Another action</a></li>
        <li><a href="#">Something else here</a></li>
        <li class="divider"></li>
        <li><a href="#">Separated link</a></li>
      </ul>
          </li>
</ul>
```

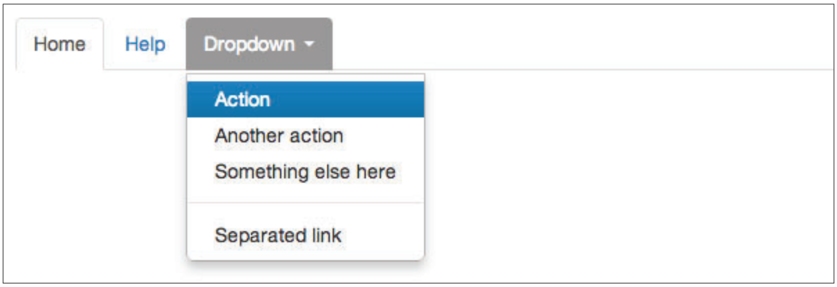



图 3-15：带下拉菜单的标签形导航

同样，如果想得到带下拉菜单的胶囊形导航（见图 3-16），只要把 .nav-tabs 类替换成 .nav-pills 即可：

```
<ul class="nav nav-pills">
        <li class="dropdown">
                <a class="dropdown-toggle" data-toggle="dropdown" href="#">
                        Dropdown
                        <b class="caret"></b>
                </a>
                <ul class="dropdown-menu">
                        <!--links-->
                </ul>
        </li>
</ul>
```

邮
电

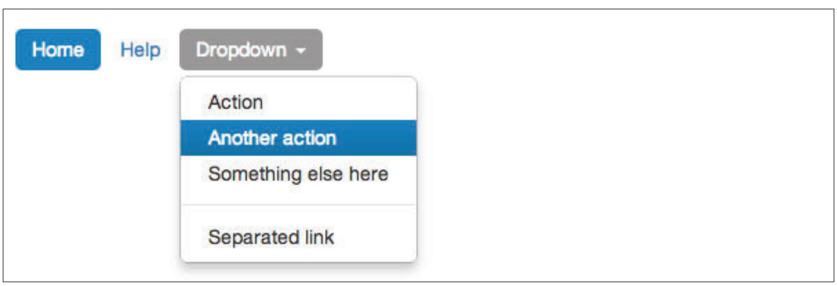



图 3-16：带下拉菜单的胶囊形导航

## 3.4.5 导航列表

导航列表可用于显示一组导航链接，在后台管理界面中的导航区比较常见。图 3-17 展示了某后台管理界面侧边栏中的一个导航列表，其

中的链接指向常用的管理页面。Bootstrap 项目的文档也使用了这种导航列表。与我们迄今为止讨论的导航相似，导航列表也是带 `.nav` 类的无序列表，再给它添加一个 `.nav-list` 类，就可以把它变成导航列表：

```
<ul class="nav nav-list">
    <li class="nav-header">List Header</li>
    <li class="active"><a href="/">Home</a></li>
    <li><a href="#">Library</a></li>
    <li><a href="#">Applications</a></li>
    <li class="nav-header">Another List Header</li>
    <li><a href="#">Profile</a></li>
    <li><a href="#">Settings</a></li>
    <li class="divider"></li>
    <li><a href="#">Help</a></li>
</ul>
```

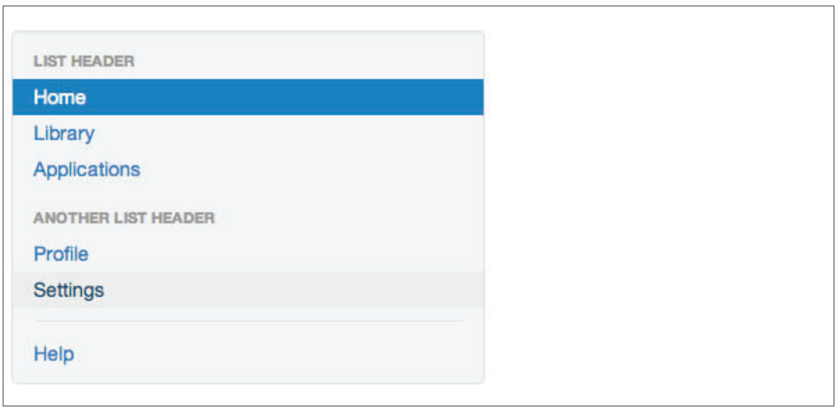



图 3-17：导航列表

## 水平分隔线

要添加水平分隔线，使用带 `.divider` 类的空的 `<li>` 元素即可：

```
<ul class="nav-menu">
    ...
        <li class="divider"></li>
        ....
</ul>
```

## 3.4.6　可切换的标签导航

除了可以创建标签形导航元素，还可以借助 JavaScript 插件添加交互功能，打开不同的标签页（见图 3-48）。一定要分别为每个标签页（.tab-pane）定义一个唯一的 ID，并把它们包装在 .tab-content 元素中：

```
<div class="tabbable">
        <ul class="nav nav-tabs">
           <li class="active"><a href="#tab1" data-toggle="tab">Meats
             </a></li>
           <li><a href="#tab2" data-toggle="tab">More Meat</a></li>
        </ul>
        <div class="tab-content">
           <div class="tab-pane active" id="tab1">
                   <p>Bacon ipsum dolor sit amet jerky flank...</p>
           </div>
           <div class="tab-pane" id="tab2">
                   <p>Beef ribs, turducken ham hock...</p>
           </div>
        </div>
</div>
```

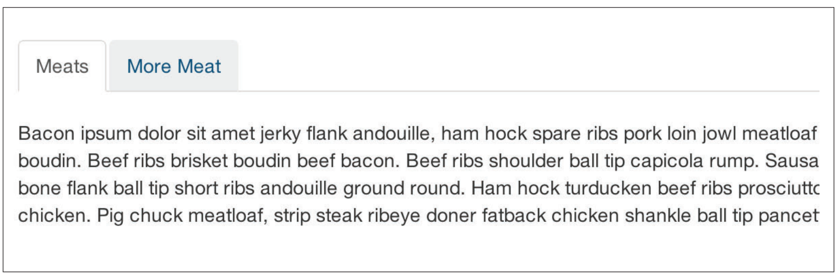



图 3-18：可切换的标签导航

如果想让标签切换时带有淡入淡出动画，只要给每个 .tab-pane 加上 .fade 类即可。

### 标签位置

标签的位置也是可以设置的，可以让它们位于标签页顶部、底部，甚至两侧（见图 3-19）。




图 3-19：标签位于底部

以下是把标签定位到底部的代码，记住 .tab-content 和 .nav-tabs 的位置也要互换：

```
<div class="tabbable tabs-below">
  <div class="tab-content">
    <div class="tab-pane active" id="tab1">
      <p>I'm in Section A.</p>
    </div>
    <div class="tab-pane" id="tab2">
      <p>I'm in Section B.</p>
    </div>
    <div class="tab-pane" id="tab3">
      <p>I'm in Section C.</p>
    </div>
  </div>
  <ul class="nav nav-tabs">
    <li class="active"><a href="#tab1" data-toggle="tab">Section A</a></li>
    <li><a href="#tab2" data-toggle="tab">Section B</a></li>
    <li><a href="#tab3" data-toggle="tab">Section C</a></li>
  </ul>
</div>
```

要让标签显示在左侧（见图 3-20），添加 .tabs-left 类：

```
<div class="tabbable tabs-left">
  <ul class="nav nav-tabs">
    <li class="active"><a href="#tab1" data-toggle="tab">Section A</a></li>
    <li><a href="#tab2" data-toggle="tab">Section B</a></li>
    <li><a href="#tab3" data-toggle="tab">Section C</a></li>
  </ul>
  <div class="tab-content">
    <div class="tab-pane active" id="tab1">
      <p>I'm in Section A.</p>
    </div>
    <div class="tab-pane" id="tab2">
      <p>I'm in Section B.</p>
    </div>
    <div class="tab-pane" id="tab3">
```

```
      <p>I'm in Section C.</p>
    </div>
  </div>
</div>
```

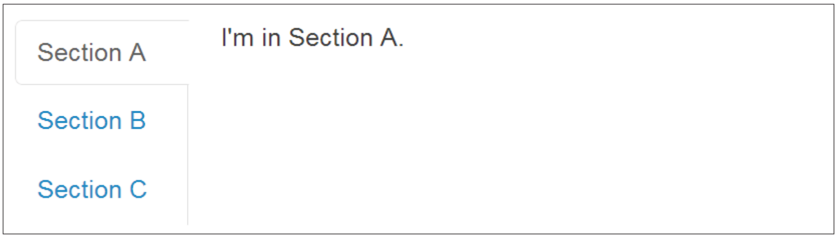



图 3-20：标签在左侧

标签在右侧要使用 `.tabs-right` 类（见图 3-21）：

```
<div class="tabbable tabs-right">
  <ul class="nav nav-tabs">
    <li class="active"><a href="#tab1" data-toggle="tab">Section A</a></li>
    <li><a href="#tab2" data-toggle="tab">Section B</a></li>
    <li><a href="#tab3" data-toggle="tab">Section C</a></li>
  </ul>
  <div class="tab-content">
    <div class="tab-pane active" id="tab1">
      <p>I'm in section A.</p>
    </div>
    <div class="tab-pane" id="tab2">
      <p>I'm in section B.</p>
    </div>
    <div class="tab-pane" id="tab3">
      <p>I'm in section C.</p>
    </div>
  </div>
</div>
```




图 3-21：标签在右侧

这里的可切换标签导航甚至可以用于设计目的之外的用途。比如，我就用它来控制导航和子导航。换句话说，单击导航菜单时，子导航会改变，显示不同的链接。

## 3.5 导航条

导航条是一个常用的组件，也是 Bootstrap 的特色功能（见图 3-22）。基本的导航条包含站点名和几个导航链接。可以对基本的导航条加以扩展，添加表单控件和下拉菜单。为保证导航条与页面同宽，可以把它放在类为 `.span12` 或者 `.container` 的容器元素中。

```
<div class="navbar">
  <div class="navbar-inner">
    <a class="brand" href="#">Title</a>
    <ul class="nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
</div>
```

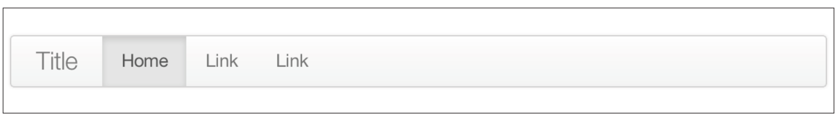



图 3-22：基本的导航条

注意代码中的 `.brand` 类，这个类会把文本的 `font-weight` 设置得更细一些，把 `font-size` 设置得更大一些。

```
<a class="brand" href="#">Project name</a>
```

### 3.5.1 导航条链接

给导航条（图 3-23）添加链接很简单，就是在导航条代码中嵌入无序列表，并为它添加 `.nav` 类。如果想在链接间添加分隔线，就添加一个类为 `.divider-vertical` 的空列表项即可：

```
<ul class="nav">
  <li class="active"><a href="#">Home</a></li>
  <li><a href="#">First Link</a></li>
  <li><a href="#">Second Link</a></li>
  <li class="divider-vertical"></li>
  <li><a href="#">Third Link</a></li>
</ul>
```




图 3-23：导航链接

## 3.5.2　表单

在导航条中嵌入表单要使用 `.navbar-form` 类，这样才能保证表单的外边距合适，而且与导航样式一致（见图 3-24）。另外，`.pull-left` 和 `.pull-right` 辅助类可以把表单定位到导航条左侧或右侧：

```
<form class="navbar-form pull-left">
        <input type="text" class="span2" id="fname">
        <button type="submit" class="btn">
</form>
```

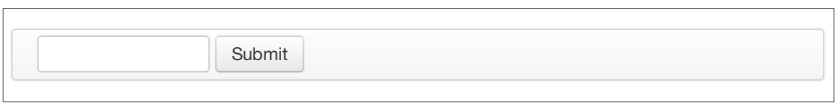



图 3-24：默认的导航条表单

想要 iOS 设备中的那种圆头输入框（见图 3-25）？用 `.navbar-search` 类代替 `.navbar-form` 类：

```
<form class="navbar-search"  accept-charset="utf-8">
        <input type="text" class="search-query" placeholder="Search">
</form>
```

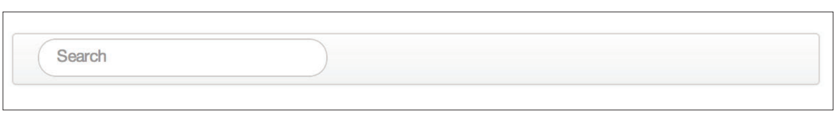



图 3-25：导航条搜索输入框

### 3.5.3 导航条菜单

Bootstrap 导航条的定位方式多种多样。默认情况下，导航条是块级元素，其定位依据是它在 HTML 中的位置。为导航条添加不同的辅助类，可以改变它的定位方式，比如让它出现在页面顶部、底部，或者随着页面滚动而滚动。

#### 1. 固顶导航条

如果想让导航条固定在页面顶部，在 .navbar 类后面再增加一个 .navbar-fixed-top 类即可。为防止导航挡住页面中的其他元素，至少要给 <body> 标签添加 40 像素的内边距：

```
<div class="navbar navbar-fixed-top">
  <div class="navbar-inner">
    <a class="brand" href="#">Title</a>
    <ul class="nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
</div>
```

#### 2. 固底导航条

要让导航条粘附于页面底部，只要把 .navbar-fixed-top 类改为 .navbar-fixed-bottom 类即可。同样，为防止导航条与其他元素重叠，至少要给 <body> 添加 40 像素的内边距：

```
<div class="navbar navbar-fixed-bottom">
 <div class="navbar-inner">
    <a class="brand" href="#">Title</a>
    <ul class="nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
</div>
```

#### 3. 静态滚动导航条

要创建一个能随页面滚动的导航条，给 .navbar 后面添加 .navbar-

static-top 类。使用这个类的时候，不要求给 <body> 添加内边距：

```
<div class="navbar navbar-static-top">
  <div class="navbar-inner">
    <a class="brand" href="#">Title</a>
    <ul class="nav">
      <li class="active"><a href="#">Home</a></li>
      <li><a href="#">Link</a></li>
      <li><a href="#">Link</a></li>
    </ul>
  </div>
</div>
```

## 4．响应式导航条

跟 Bootstrap 中的其他组件一样，导航条也完全可以像图 3-26 所示的那样具有响应能力。想让导航条具有响应能力，请把需要隐藏的内容包装到类为 .nav-collapse.collapse 的 <div> 中。（请参考下面的代码。）显示隐藏内容的按钮（链接）要有一个 .btn-navbar 类和两个 data-* 属性：data-toggle 会告诉 JavaScript 这个按钮要做什么，data-target 表示要切换的元素。三个类为 .icon-bar 的 <span> 可以创建“汉堡式按钮”，用于切换类为 .nav-collapse 的 <div> 中的元素。要实现响应式导航条，还必须包含 bootstrap-responsive.css 和 collapse.js 或完整的 bootstrap.js 文件。




图 3-26：响应式导航条

以下是创建响应式导航条的示例代码：

```
<div class="header">
  <div class="navbar-inner">
    <div class="container">
      <a class="btn btn-navbar" data-toggle="collapse"
                                data-target=".nav-collapse">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </a>

      <!-- 始终显示导航条的名字 -->
      <a href="#" class="brand">Project Name</a>

      <!-- 把所有需要隐藏的项目添加到这里 -->
      <div class="nav-collapse collapse">
        <!-- .nav、.navbar-search 等等 -->
      </div>

    </div>
  </div>
</div>
```

### 5. 反色导航条

要创建反色导航条，即图 3-27 所示的黑底白字的导航条，只要在 `.navbar` 类后面添加 `.navbar-inverse` 类即可：

```
<div class="navbar navbar-inverse">
        ...
</div>
```

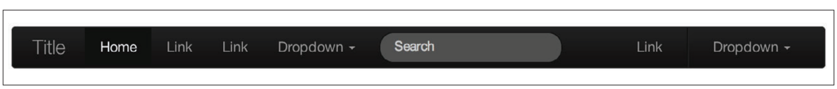



图 3-27：反色导航条

## 3.6　面包屑式导航

面包屑式导航非常适合展示站点的层级关系（见图 3-28）。以博客为例，面包屑式导航可以展示发表日期、类别或标签。Bootstrap 中的面包屑式导航就是一个类为 `.breadcrumb` 的无序列表。还有一个辅助类

.divider 能减淡颜色、减小字号，利用这个类可以让斜杠、箭头等符号变成分隔符。注意，从标记来看，面包屑式导航中的分隔符与导航条中的不太一样。

以下代码使用 .breadcrumb 创建面包屑式导航：

```
<ul class="breadcrumb">
    <li><a href="#">Home</a> <span class="divider">/</span></li>
    <li><a href="#">2012</a> <span class="divider">/</span></li>
    <li><a href="#">December</a> <span class="divider">/</span></li>
    <li><a href="#">5</a></li>
</ul>

<ul class="breadcrumb">
  <li><a href="#">Home</a> <span class="divider">&rarr;</span></li>
  <li><a href="#">Dinner Menu</a> <span class="divider">&rarr;</span></li>
  <li><a href="#">Specials</a> <span class="divider">&rarr;</span></li>
  <li><a href="#">Steaks</a></li>
</ul>

<ul class="breadcrumb">
  <li><a href="#">Home</a> <span class="divider">&raquo;</span></li>
  <li><a href="#">Electronics</a> <span class="divider">&raquo;</span></li>
  <li><a href="#">Raspberry Pi</a></li>
</ul>
```




图 3-28：面包屑式导航

## 3.7　分页导航

Bootstrap 分页导航使用的界面元素同样是无序列表，而且这个无序列表同样包含在以一个特殊的类区别于其他元素的 <div> 中。最简单的情况下，给 <div> 添加 .pagination 类，就可以创建一行带边框的链

接（见图 3-29）。无序列表中的每个列表项都可以使用 .disabled 或 .active 类进一步修饰。图 3-30 展示了使用这两个辅助类的例子。

以下是简单的分页导航：

```
<div class="pagination">
  <ul>
    <li><a href="#">&laquo;</a></li>
    <li><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">&raquo;</a></li>
  </ul>
</div>
```

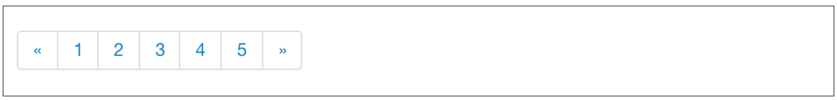



图 3-29：简单的分页导航

以下是使用辅助类进一步修饰后的分页导航：

```
<div class="pagination pagination-centered">
  <ul>
    <li class="disabled"><a href="#">«</a></li>
    <li class="active"><a href="#">1</a></li>
    <li><a href="#">2</a></li>
    <li><a href="#">3</a></li>
    <li><a href="#">4</a></li>
    <li><a href="#">5</a></li>
    <li><a href="#">»</a></li>
  </ul>
</div>
```

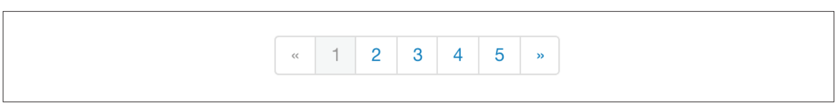



图 3-30：使用辅助类的分页导航

除了给列表项添加 .active 和 .disabled 类，还可以给它们的容器 <div> 添加 .pagination-centered 类，从而使分页导航居中。类似地，

要使分页导航与容器右对齐，则使用 .pagination-right 类。此外，还可以调整分页导航的大小，用于调整大小的类有 3 个：.pagination-large、.pagination-small 和 .pagination-mini，代码如下，效果见图 3-31：

```
<div class="pagination pagination-large">
  <ul>
    ...
  </ul>
</div>
<div class="pagination">
  <ul>
    ...
  </ul>
</div>
<div class="pagination pagination-small">
  <ul>
    ...
  </ul>
</div>
<div class="pagination pagination-mini">
  <ul>
    ...
  </ul>
</div>
```

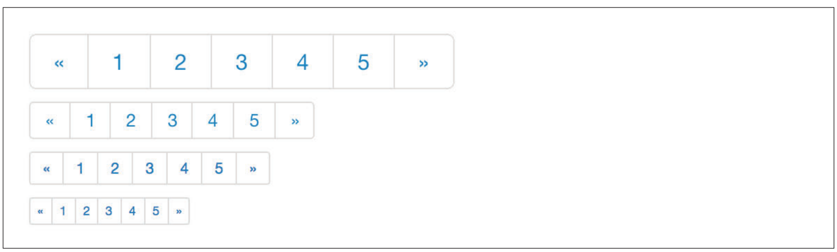



图 3-31：调整分页导航的大小

## 前后页导航

如果你想创建更简单的分页导航，可以使用前后页导航。前后页导航与分页导航类似，也需要把一个无序列表包装在一个 <div> 中。默认情况下，前后页导航链接是居中的（见图 3-32）。

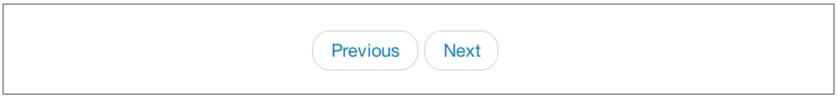



图 3-32：基本的前后页导航

以下是实现前后页导航的代码：

```
<ul class="pager">
  <li><a href="#">Previous</a></li>
  <li><a href="#">Next</a></li>
</ul>
```

要让导航链接与页面左对齐或右对齐，需要给相应的列表项添加 `.previous` 和 `.next` 类（见图 3-33）。与图 3-30 中的分页导航一样，也可以通过添加 `.disabled` 类创建灰色禁用按钮的效果。




图 3-33：对齐后的页面链接

以下是对齐页面链接的代码：

```
<ul class="pager">
  <li class="previous">
    <a href="#">&larr; Older</a>
  </li>
  <li class="next">
    <a href="#">Newer &rarr;</a>
  </li>
</ul>
```

# 3.8　标签

标签非常适合显示计数结果、提示信息等内容。Bootstrap 提供的各种标签我都非常喜欢，图 3-34 展示了这些标签的效果。




图 3-34：各种标签

以下是实现这些标签的代码：

```
<span class="label">Default</span>
<span class="label label-success">Success</span>
<span class="label label-warning">Warning</span>
<span class="label label-important">Important</span>
<span class="label label-info">Info</span>
<span class="label label-inverse">Inverse</span>
```

## 3.9 圆标签

圆标签（badge）与标签类似，主要区别在于它们的两头更圆。圆标签的颜色与标签的颜色是一样的（见图 3-35）。




图 3-35：圆标签

以下代码展示了如何使用圆标签：

```
<span class="badge">1</span>
<span class="badge badge-success">2</span>
<span class="badge badge-warning">4</span>
<span class="badge badge-important">6</span>
<span class="badge badge-info">8</span>
<span class="badge badge-inverse">10</span>
```

## 3.10 排版相关的元素

除了按钮、标签、表单、表格、标签页，Bootstrap 还提供了一些用于页面排版的元素。

### 3.10.1 重头消息

重头消息（hero-unit）是一个内容区块，这个区块中的标题字号是加大的，而且边距也比较大，适合放在着陆页面（见图 3-36）。要在页面中放一个重头消息，只要给容器 `<div>` 添加 `.hero-unit` 类即可。除了 `<h1>` 的字号会加大以外，所有文本的 `font-weight` 都被减少到了 200：

```
<div class="hero-unit">
  <h1>Hello, World!</h1>
  <p>This is a simple hero unit, a simple jumbotron-style component for
  calling extra attention to featured content or information.</p>
  <p><a class="btn btn-primary btn-large">Learn more</a></p>
</div>
```

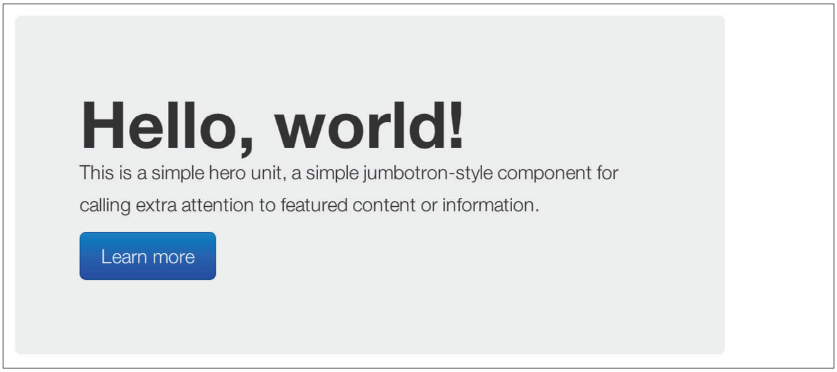



图 3-36：重头消息

### 3.10.2　页面标题

页面标题（见图 3-37）会给页面中的标题添加更多的间隙，非常适合区分一个页面中的多个标题。要使用页面标题，把标题都包装在类为 `.page-header` 的 `<div>` 中即可：

```
<div class="page-header">
  <h1>Example page header <small>Subtext for header</small></h1>
</div>
```




图 3-37：页级标题

## 3.11　缩略图

很多网站都需要以网格布局显示图片，Bootstrap 为此提供了一种便捷的组件——缩略图。要创建缩略图，只要给图片加上 `<a>` 标签，再给

这个标签添加 .thumbnail 类即可。这样就可以给图片加上 4 像素的内边距和一个灰色边框（见图 3-38）。而且，鼠标指针悬停时，图片四周还会出现动态光晕。




图 3-38：简单的缩略图

使用以下代码创建缩略图：

```
<a href="#" class="thumbnail">
  <img alt="Kittens!" style="" src="http://placekitten.com/300/250">
</a>
```

在这个简单的缩略图基础上，还可以添加标题、按钮、文本，如图 3-39 所示。为此，需要把类为 .thumbnail 的 <a> 标签换成 <div>。在这个 <div> 中，就可以添加任何内容了。由于此时的容器是 <div>，因此可以通过默认的 .span* 类来设置缩略图大小。如果要显示多个缩略图，那就把它们放到一个无序列表中，每个列表项都会浮动到左侧。

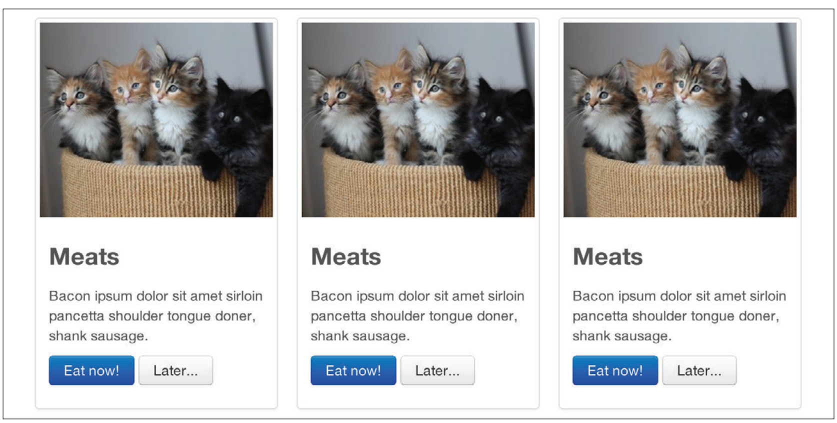



图 3-39：包含更多内容的缩略图

以下是为缩略图添加更多内容的示例代码：

```
<ul class="thumbnails">
  <li class="span4">
    <div class="thumbnail">
      <img data-src="holder.js/300x200" alt="300x200" style="">
      <div class="caption">
        <h3>Meats</h3>
        <p>Bacon ipsum dolor sit amet sirloin pancetta shoulder tongue doner,
           shank sausage.</p>
        <p><a href="#" class="btn btn-primary">Eat now!</a> <a href="#"
           class="btn">Later...</a></p>
      </div>
    </div>
  </li>
  <li class="span4">
    ...
  </li>
</ul>
```

# 3.12　警示框

与第 4 章会讲到的模态对话框类似，警示框是一种为用户消息添加样式的组件（见图 3-40）。默认的警示框效果可以通过给 <div> 添加 .alert 类来实现：

```
<div class="alert">
    <a href="#" class="close" data-dismiss="alert">&times;</a>
    <strong>Warning!</strong> Not to be alarmist, but you have now been alerted.
</div>
```

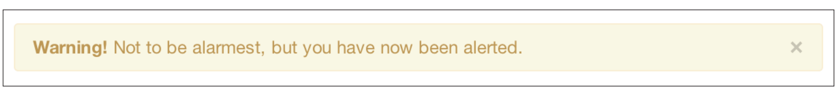



图 3-40：简单的警示框

这个 .alert 使用的是 jQuery 的警示框插件，要显示关闭警示框按钮，可以创建一个包含 data-dismiss="alert" 属性的按钮。移动版 Safari 和 Opera 浏览器还需要加上 href="#" 才行。

如果用户消息太长，可以考虑使用 .alert-block 类。这个类会给包含消息的容器上、下多添加一些边距，让多行文本看起来更舒服（见图 3-41）。

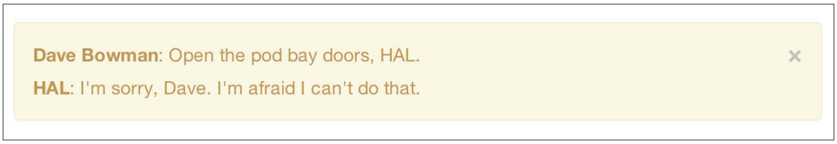



图 3-41：警示信息块

如图 3-42 所示，还有三种颜色选项，可以帮助我们传达警示的含义。这三种颜色的警示框分别需要额外添加 .alert-error、.alert-success 和 .alert-info 类。

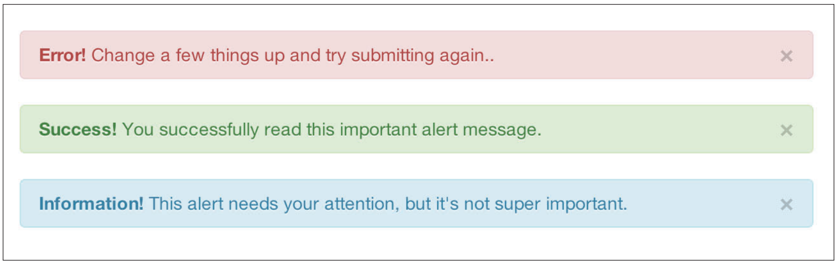



图 3-42：警示框的颜色

## 3.13 进度条

进度条用于显示资源下载进度，或者表示页面中的元素发生了状况。个人认为，进度条在 Bootstrap 之外没有什么用武之地。换句话说，在使用 Bootstrap 的芸芸众生中，真会创建进度条的人只是极少数。自然地，进度条本身只是一些静态元素，需要 JavaScript 代码来使其拥有交互能力。

默认的进度条背景为浅灰色，进度条本身为蓝色，如图 3-43 所示。进度条背景用类为 .progress 的 <div> 创建，进度条用类为 .bar 的 <div> 创建。进度条的百分比则是通过在 style 属性中指定 CSS 的 width 值设置的。这里的 style="width: 60%;" 表示进度完成了 60%：

```
<div class="progress">
  <div class="bar" style="width: 60%;"></div>
</div>
```



图 3-43：默认的进度条

创建带条纹的进度条（见图 3-44）[1]，只需给容器 `<div>` 添加 `.progress-striped` 类即可：

```
<div class="progress progress-striped">
  <div class="bar" style="width: 20%;"></div>
</div>
```

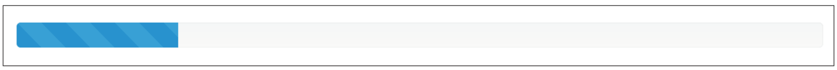


图 3-44：带条纹的进度条

这个带条纹的进度条是静态的。还可以让条纹动起来，像理发店门口的转花筒一样（图 3-45 是动态条纹进度条的静态截图）。



图 3-45：动态条纹版的进度条

以下是让进度条动起来的代码：

```
<div class="progress progress-striped active">
  <div class="bar" style="width: 40%;"></div>
</div>
```

除了蓝色进度条之外，还可以有绿色、黄色和红色进度条，需要额外添加的类分别是 `.bar-success`、`.bar-warning` 和 `.bar-danger`。还可像下面这样把进度条拼接起来，基于多个元素构成某种图案：

```
<div class="progress">
  <div class="bar bar-success" style="width: 35%;"></div>
  <div class="bar bar-warning" style="width: 20%;"></div>
  <div class="bar bar-danger" style="width: 10%;"></div>
</div>
```

注 1：条纹式进度条在 IE7 和 IE8 中无效。



图 3-46：叠加起来的进度条

## 3.14 媒体对象

在 Facebook、Twitter 等社交网站上，去掉一些与时间线相关的格式，就能看到媒体对象了（见图 3-47）。Bootstrap 根据 OOCSS（面向对象的 CSS）社区的共识推出的媒体对象，其目的就是为了减少开发这种信息块的代码。Nicole Sullivan-Hass 在她自己的站点（http://www.stubbornella.org/content/2010/06/25/the-media-object-saves-hundreds-of-lines-of-code/）上分享了一些与 Bootstrap 提供的类似的媒体对象。有了媒体对象，不仅能够节省数百行代码，还可简化自定义。




图 3-47：媒体对象

Bootstrap 把设计和添加样式的自由留给了用户，但为此提供了坚实的基础。与 Bootstrap 中的其他组件一样，媒体对象的目标（标记少、易扩展）也是通过给一些简单的标记应用类来实现的。有两种媒体对象：`.media` 和 `.media-list`。图 3-48 展示了前一种形式的媒体对象。如果你准备的是一个无序列表，那么就使用 `.media-list`；如果你只有一个 `<div>` 元素，那么就使用 `.media`：

```
<div class="media">
  <a class="pull-left" href="#">
    <img class="media-object" data-src="holder.js/64x64">
  </a>
  <div class="media-body">
    <h4 class="media-heading">Media heading</h4>
    <p>...</p>

    <!-- 嵌套的媒体对象 -->
    <div class="media">
      ...
    </div>
  </div>
</div>
```

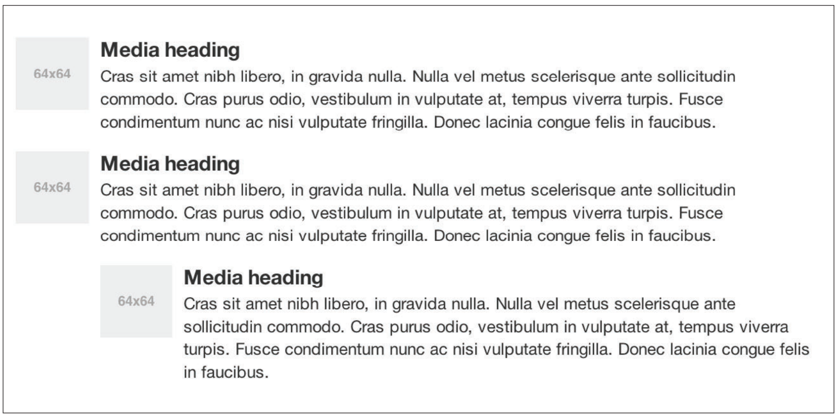



图 3-48：默认的媒体对象

要使用媒体列表（见图 3-49），把容器 `<div>` 改为 `<ul>`，然后添加 `.media-list` 类。因为媒体对象可以嵌套，所以很适合展示评论或其他需要以列表展示的信息。




图 3-49：媒体列表示例

以下是创建媒体列表的代码：

```
<ul class="media-list">
  <li class="media">
    <a class="pull-left" href="#">
      <img class="media-object" data-src="holder.js/64x64">
    </a>
    <div class="media-body">
      <h4 class="media-heading">Media heading</h4>
      <p>...</p>
      ...

      <!-- 嵌套的媒体对象 -->
      <div class="media">
        ...
     </div>
    </div>
  </li>
</ul>
```

## 3.15　其他组件

还有几个 Bootstrap 组件本章没有介绍，有的是用于布局的，有的只是辅助类。比如，洼地（well）。

### 3.15.1　洼地

洼地（well）就是一个容器 `<div>`，让其中的内容看起来像是陷入页面一样（见图 3-50）。我一般用洼地来显示博客的元信息，比如作者、日期和类别。要创建洼地，只要把内容包装在类为 `.well` 的 `<div>` 中即可：

```
<div class="well">
  ...
</div>
```

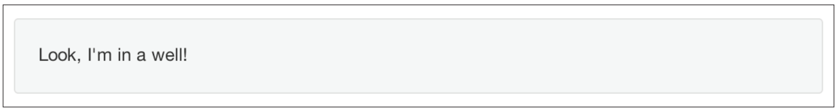



图 3-50：洼地

还有两个类可以与 `.well` 同时使用：`.well-large` 和 `.well-small`。这两个类影响内边距，前者会增大内边距，后面会减小内边距（见图 3-51）。




图 3-51：与洼地相关的可选类

以下代码使用了洼地相关的类：

```
<div class="well well-large">

  Look, I'm in a .well-large!

</div>

<div class="well well-small">

  Look, I'm in a .well-small!

</div>
```

## 3.15.2　辅助类

以下是几个方便的辅助类。

### 1．向左浮动

要想把元素浮动到容器左侧，使用 .pull-left 类：

```
<div class="pull-left">
  ...
</div>

.pull-left {
  float: left;
}
```

### 2．向右左浮动

要想把元素浮动到容器右侧，使用 .pull-right 类：

```
<div class="pull-right">
  ...
</div>

.pull-right {
  float: right;
}
```

### 3．清除浮动

要清除任何元素的浮动，使用 .clearfix 类。如果有两个不同大小的元素都向一侧浮动，就需要把那个在代码中位于后面的元素向下推或者说清除之前的内容。使用一个带 .clearfix 类的空 <div> 就可以达到这个目的：

```
<div class="clearfix"></div>

.clearfix {
  *zoom: 1;
  &:before,
  &:after {
    display: table;
    content: "";
  }
  &:after {
    clear: both;
  }
}
```

第 4 章

# Bootstrap支持的JavaScript插件

第 3 章介绍的组件只是个开始。在此基础上，Bootstrap 还附带了 13 个 jQuery 插件，用于扩展网站的功能、丰富用户体验。要使用 Bootstrap 的 JavaScript 插件不必非得是 JavaScript 高手。事实上，利用 Bootstrap 的 Data API，大多数插件无需编写一行代码就可以触发。

## 4.1 概述

Bootstrap 有两种形式，一种是编译后的，一种是原始格式。Bootstrap 2.2.2 未经压缩的版本是 59 KB，而压缩之后的版本只有 32 KB。所有 Bootstrap 的插件都可以使用内置的 Data API 访问。有了 Data API，不需要写 JavaScript 代码，就能为网页添加插件。

一般来说，JavaScript 代码都保存在单独的文件中，或者放在页面 `</body>` 标签之前。如果是单独的文件，要使用 `<script>` 的 `src` 属性把它链接到网页中。如果不是单独的文件，则可以把代码写在 `<script>...</script>` 标签内部：

```
<!-- 引用单独的 JavaScript 文件 -->
<script src="assets/js/javaScript.js"></script>

<!-- 在网页中直接写 JavaScript -->
<script type="text/javaScript">
        function js_alert{
                alert('Page has loaded');
        }
</script>
```

通常，最好先检查确认 DOM 已经就绪，然后再执行 JavaScript 调用语句。在 DOM 就绪前执行 JavaScript 调用的话，浏览器可能还没有准备好要操作的元素。使用 jQuery，先选择文档（或整个页面的内容）然后调用 `.ready()` 方法就可以了：

```
$(document).ready(function(){
        alert('Page has loaded');
        // 页面加载完成后，才会弹出警告框
});
```

如前所述，Bootstrap 提供了可以嵌入 HTML 元素中作为数据属性的

Data API。如果想解除对数据属性的绑定，可以使用以下 JavaScript 代码：

```
$('body').off('.data-api')
```

如果你只想禁用某一个插件，可以使用这个插件的命名空间加上 `data-api` 命名空间：

```
$('body').off('.alert.data-api')
```

### 可编程的API

Bootstrap 的开发者希望用户通过 JavaScript API 使用所有插件。所有公开的 API 都是单个可连缀的方法，返回操作的元素集合：

```
$('.btn.danger').button('toggle').addClass('active')
```

所有方法都可以接受一个可选的选项对象、一个表示要调用的方法的字符串，或者什么参数也不接受（表示初始化插件的默认行为）：

```
$("#myModal").modal()   // 初始化默认功能
$("#myModal").modal({ keyboard: false })     // 初始化不支持键盘
$("#myModal").modal('show')                  // 初始化并立即调用 show
```

## 4.2　过渡

过渡插件可以实现简单的过渡效果，比如：

- 模态对话框的滑入滑出和淡入淡出；
- 标签页淡出；
- 警告框淡出；
- 滑入滑出旋转面板。

## 4.3　模态框

模态框就是一个叠放在父窗口上的子窗口（见图 4-1）。模态框经常用于显示来自其他地方的内容，让人可以与之交互但又不脱离当前窗口

的上下文。这种子窗口可以展示信息、提供交互，等等。我一般使用模态框来盛放幻灯片或登录 / 注册信息。这个插件差不多是我喜欢的 Bootstrap 插件了。

要创建静态的模态框，使用以下代码：

```
<div class="modal hide fade">
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal" aria-hidden=
      "true">&times;</button>
    <h3>Modal header</h3>
  </div>
  <div class="modal-body">
    <p>One fine body...</p>
  </div>
  <div class="modal-footer">
    <a href="#" class="btn">Close</a>
    <a href="#" class="btn btn-primary">Save changes</a>
  </div>
</div>
```

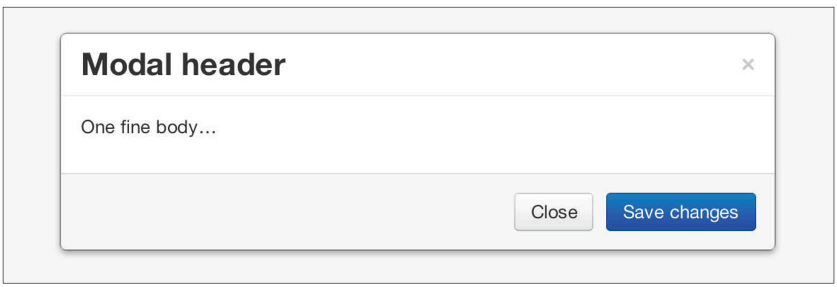



图 4-1：静态的模态框

要打开模态框，必须有一个触发装置。我一般会使用按钮或链接，在下面代码的 `<a>` 标签中，`href="myModal"` 表示你要在页面上加载的模态框。可以在同一个页面上创建多个模态框，并为它们都准备一个触发装置。当然，多个模态框不一定同时都加载到页面中，可以根据需要临时向页面中加载。

关于模态框，重点要关注 3 个类。第一个是 `.modal`，它只用来把一个 `<div>` 标注为模态框。第二个是 `.hide`，这个类告诉浏览器先把模态框

隐藏起来，在用户单击触发装置时再显示。最后一个是 .fade，这个类会导致模态框从无到有或从有到无时，以淡入淡出的效果呈现。

```
<!-- 触发模态框的按钮 -->
<a href="#myModal" role="button" class="btn" data-toggle="modal">
  Launch demo modal</a>

<!-- 模态框 -->
<div id="myModal" class="modal hide fade" tabindex="-1" role="dialog"
aria-labelledby="myModalLabel" aria-hidden="true">
  <div class="modal-header">
    <button type="button" class="close" data-dismiss="modal"
    aria-hidden="true">×</button>
    <h3 id="myModalLabel">Modal header</h3>
  </div>
  <div class="modal-body">
    <p>One fine body...</p>
  </div>
  <div class="modal-footer">
    <button class="btn" data-dismiss="modal"
    aria-hidden="true">Close</button>
    <button class="btn btn-primary">Save changes</button>
  </div>
</div>
```

## 4.3.1 用法

使用 Bootstrap 的 JavaScript Data API，只要设置几个数据属性，就可实现模态框的调用。为此，在作为开关的链接或按钮上设置 `data-toggle="modal"`，再把 `data-target="#foo"` 设置为模态框的 ID。

要通过 JavaScript 调用 `id="myModal"` 的模态框，只需一行代码：

```
$('#myModal').modal(options)
```

## 4.3.2 选项

选项可以通过数据属性传入也可以通过 JavaScript 传入。要使用数据属性，请在选项名称前加上 `data-`（如 `data-backdrop=""`）。表 4-1 列出了模态框的一些选项及说明。

表4-1　模态框的选项

| 名　称 | 类　型 | 默认值 | 说　　明 |
| --- | --- | --- | --- |
| backdrop | Boolean | `true` | 包含模态框背板元素。如果不想单击背板关闭模态框，为这个数据属性指定 `static` |
| keyboard | Boolean | `true` | 按 Esc 键关闭模态框 |
| show | Boolean | `true` | 初始状态下模态框是否可见 |
| remote | URL | `false` | 如果提供了远程 URL，内容将通过 jQuery 的 `load` 方法加载，然后插入 `.modal-body` 元素。如果使用的是数据 API，可以使用 `href` 属性指定远程资源地址，下面是一个例子：<br>`<a data-toggle="modal" href="remote.html" data-target="#modal">click me</a>` |

## 4.3.3　方法

下面是模态框相关的几个方法。

### 1. 选项

将内容激活为模态框，接受可选的选项对象。

`.modal(options).`

```
$('#myModal').modal({
        keyboard: false
})
```

### 2. 开关

手动开关模态框。

`.modal('toggle').`

```
$('#myModal').modal('toggle')
```

### 3. 显示

手动打开模态框。

`.modal('show').`

```
$('#myModal').modal('show')
```

### 4. 关闭

手动关闭模态框。

```
.modal('hide').
    $('#myModal').modal('hide')
```

## 4.3.4 事件

Bootstrap 提供了表 4-2 所示的事件，可用于调用函数。

表4-2 模态框事件

| 事 件 | 说 明 |
|---|---|
| show | 在实例方法 `show` 被调用时触发 |
| shown | 在模态框已经对用户可见时（CSS 过渡完成后）触发 |
| hide | 在实例方法 `hide` 被调用时触发 |
| hidden | 在模态框已经对用户不可见时（CSS 过渡完成后）触发 |

下面这个例子是在模态框关闭后弹出一个警告框：

```
$('#myModal').on('hidden', function () {
        alert('Hey girl, I heard you like modals...');
})
```

# 4.4 下拉菜单

第 3 章详细介绍过下拉菜单，但并没有提到下拉菜单的交互功能。我们知道，下拉菜单可以放在导航条、胶囊形导航、标签形导航和按钮导航中。

## 4.4.1 用法

要使用下拉菜单（见图 4-2），给一个链接或按钮添加 `data-toggle="dropdown"` 属性，以便能够通过它打开下拉菜单。

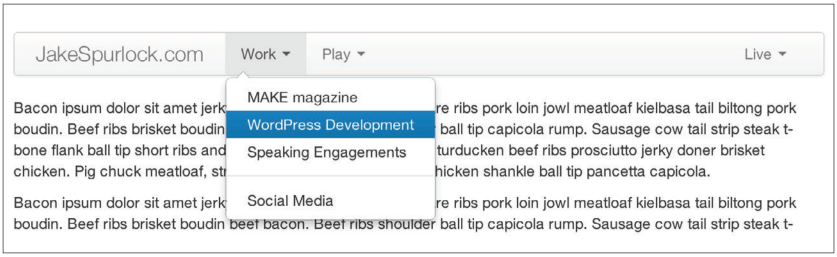



图 4-2：导航条中的下拉菜单

以下是创建带有数据属性的下拉菜单的代码：

```
<li class="dropdown">
  <a href="#" id="drop" role="button" class="dropdown-toggle"
  data-toggle="dropdown">Word <b class="caret"></b></a>
  <ul class="dropdown-menu" role="menu" aria-labelledby="drop">
    <li><a tabindex="-1" href="#">MAKE magazine</a></li>
    <li><a tabindex="-1" href="#">WordPress DevelopmentS</a></li>
    <li><a tabindex="-1" href="#">Speaking Engagements</a></li>
    <li class="divider"></li>
    <li><a tabindex="-1" href="#">Social Media</a></li>
  </ul>
</li>
```

如果你想让链接任何时候都可以打开（比如在 JavaScript 无效时也能打开），就用 `data-target="#"` 代替 `href="#"`，并给 `href` 属性指定链接：

```
<div class="dropdown">
  <a class="dropdown-toggle" id="dLabel" role="button"
  data-toggle="dropdown" data-target="#" href="/page.html">
    Dropdown
    <b class="caret"></b>
  </a>
  <ul class="dropdown-menu" role="menu" aria-labelledby="dLabel">
    ...
  </ul>
</div>
```

## 4.4.2 通过JavaScript使用下拉菜单

要通过 JavaScript 调用下拉菜单，使用如下代码：

```
$('.dropdown-toggle').dropdown()
```

### 4.4.3　方法

有一个简单的方法可以在代码里打开或隐藏下拉菜单，没有选项：

```
$().dropdown('toggle')
```

## 4.5　滚动监控器

Scrollspy 插件（见图 4-3）可以根据滚动位置更新导航链接。在静态实现中，可以根据滚动位置给导航条添加 .active 类。要通过数据属性添加 Scrollspy 插件，把 data-spy="scroll" 添加给要“监控”（spy）的元素（通常是页面主体），把 data-target=".navbar" 设置为要根据滚动改变类的滚动条。另外，被监控的元素中还要有对应的元素，其 ID 与导航条链接的引用目标一致。

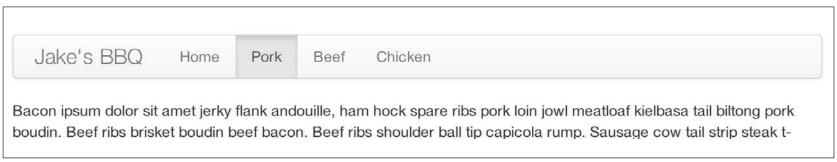



图 4-3：滚动监控示例

### 4.5.1　用法

如果要监控页面主体，需要给 <body> 标签添加 data-spy= "scroll"，以及一个 data-target=".navbar" 属性，当然这个 data-target 指向显示导航链接的导航条：

```
<body data-spy="scroll" data-target=".navbar">...</body>
```

在导航条中，需要让每个链接都充当页面中锚点（ID）的指示器（href）：

```
<div class="navbar">
  <div class="navbar-inner">
    <div class="container">
      <a class="brand" href="#">Jake's BBQ</a>
      <div class="nav-collapse">
```

```
            <ul class="nav">
              <li class="active"><a href="#">Home</a></li>
              <li><a href="#pork">Pork</a></li>
              <li><a href="#beef">Beef</a></li>
              <li><a href="#chicken">Chicken</a></li>
            </ul>
          </div><!-- /.nav-collapse -->
        </div>
      </div><!-- /navbar-inner -->
    </div>
```

### 通过JavaScript使用

如果不是通过数据属性，而是要通过 JavaScript 来调用滚动监控插件，首先要选择作为指示器的滚动条，然后调用 `.scrollspy()` 函数：

```
$('#navbar').scrollspy()
```

## 4.5.2　.scrollspy('refresh')方法

在通过 JavaScript 调用滚动监控插件时，需要调用 `.scrollspy('refresh')` 方法更新 DOM。这样可以反映最新 DOM 结构。

```
$('[data-spy="scroll"]').each(function () {
        var $spy = $(this).scrollspy('refresh')
});
```

## 4.5.3　选项

选项可以通过数据属性或 JavaScript 设置。在使用数据属性时，要给选项名称前面加上 `data-` 前缀，比如 `data-offset=""`（见表 4-3）。

表4-3　滚动监控器选项

| 名　称 | 类　型 | 默认值 | 说　明 |
| --- | --- | --- | --- |
| offset | number | 10 | 计算滚动位置时距离顶部的偏移量 |

在使用固定导航条的情况下，`offset` 选项是很有用的。为正确计算滚动位置，应该将这个选项设置为 50 像素左右（见表 4-4）。

### 4.5.4 事件

表4-4 滚动监控器事件

| 事 件 | 说 明 |
| --- | --- |
| activate | 滚动监控器激活新项时触发 |

## 4.6 可切换的标签页

第 3 章介绍了可切换的标签导航。通过组合一些数据属性，很容易创建标签页式界面（见图 4-4）。为此，先创建一个 .nav 用于导航，然后再把标签页的内容包装在类为 .tab-content 的 <div> 中：

```
<ul class="nav nav-tabs">
    <li><a href="#home" data-toggle="tab">Home</a></li>
        <li><a href="#profile" data-toggle="tab">Profile</a></li>
        <li><a href="#messages" data-toggle="tab">Messages</a></li>
        <li><a href="#settings" data-toggle="tab">Settings</a></li>
</ul>

<div class="tab-content">
        <div class="tab-pane active" id="home">...</div>
        <div class="tab-pane" id="profile">...</div>
        <div class="tab-pane" id="messages">...</div>
        <div class="tab-pane" id="settings">...</div>
</div>
```

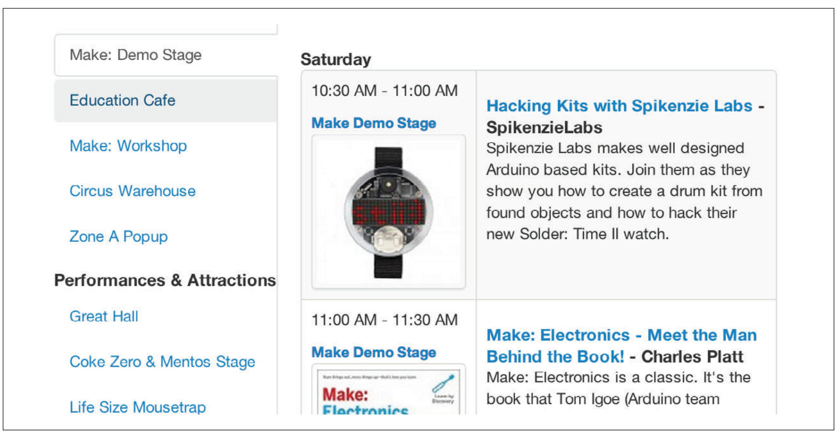



图 4-4：可切换的标签页

## 4.6.1 用法

要在页面中启用标签页，可以使用 Bootstrap 的 Data API，也可以直接使用 JavaScript。使用 Data API 的情况下，需要给链接添加 `data-toggle` 属性，然后就可以按照链接的 `href` 属性，激活带 `.tab-pane` 类和相关 ID 的元素。此外，用 `data-target=""` 取代 `href="#"` 也可以实现同样的功能。

以下是通过 JavaScript 激活标签页的代码：

```
$('#myTab a').click(function (e) {
       e.preventDefault();
       $(this).tab('show');
})
```

以下是切换到个别标签页的不同方式：

```
$('#myTab a[href="#profile"]').tab('show'); // 按照名称选择标签页
$('#myTab a:first').tab('show'); // 选择第一个标签页
$('#myTab a:last').tab('show'); // 选择最后一个标签页
$('#myTab li:eq(2) a').tab('show'); // 选择第三个标签页（索引 0 为第一个）
```

## 4.6.2 事件

标签页有两个事件，如表 4-5 所示。

表4-5　可切换的标签页事件

| 事　件 | 说　　明 |
| --- | --- |
| show | 在标签显示但显示完之前触发。使用 `event.target` 和 `event.relatedTarget` 可以取得当前和之前活动的标签页 |
| shown | 在标签显示且显示完之后触发。使用 `event.target` 和 `event.relatedTarget` 可以取得当前和之前活动的标签页 |

以下是使用 `shown` 事件的例子：

```
$('a[data-toggle="tab"]').on('shown', function (e) {
        e.target // 当前活动的标签页
        e.relatedTarget // 之前活动标签页
})
```

要了解 .on 方法的细节，请参考 jQuery 的网站（http://api.jquery.com/on/）。

## 4.7 提示条

提示条（见图 4-5）可以用来给出链接说明或（与标签连用）给出缩写词的的全称。这个插件最初的原型是 Jason Frame 的 jQuery.tipsy 插件。但目前这个提示条利用 Bootstrap JavaScript API 可以不使用图片，而且有 CSS 动画。




图 4-5：提示条的位置

### 4.7.1 用法

给 <a> 元素添加 rel="tooltip" 就可以添加提示条，<a> 元素的 title 属性值就是提示条的内容。下面两个例子分别使用 Bootstrap Data API 和 JavaScript 来创建提示条：

```
<a href="#" rel="tooltip" title="This is the tooltip">Tooltip Example</a>
$('#example').tooltip(options)
```

### 4.7.2 选项

与其他插件一样，提示条也可以通过 Data API 或 JavaScript 来设置。只不过在使用数据属性时，选项名称前面要加上 data-，比如 title 就要变成 data-title（见表 4-6）。

表4-6　提示条选项

| 名　称 | 类　型 | 默认值 | 说　　明 |
| --- | --- | --- | --- |
| animation | Boolean | `true` | 为提示条应用 CSS 的淡入淡出过渡 |
| html | Boolean | `false` | 提示条支持插入 HTML。如果值为 `false`，则会使用 jQuery 的 `text` 方法。如果担心 XSS 攻击，那就使用文本吧 |
| placement | String \| Function | `'top'` | 设置提示条的位置：`top \| bottom \| left \| right` |
| selector | String | `false` | 如果提供了选择符，则将提示条委托给指定的目标 |
| title | String \| Function | `''` | 如果 `title` 属性不存在，则使用这个默认值 |
| trigger | String | `'hover focus'` | 设置如何触发提示条：`click \| hover \| focus \| manual`。可以传入多个触发方式，以空格分隔即可 |
| delay | Number \| Object | `0` | 延迟显示和隐藏提示条的时间（毫秒）——手动触发提示条时无效<br>如果只指定一个数值，则隐藏和显示都使用该数值<br>对象的结构为 `delay: { show: 500, hide: 100 }` |
| container | String | `false` | 将提示条附加到特定元素，如 `container: 'body'` |

## 4.7.3　方法

以下是与提示条相关的方法。

### 1. 选项

给一组元素添加提示条：

```
$().tooltip(options)
```

### 2. 显示

显示元素的提示条：

```
$('#element').tooltip('show')
```

### 3．隐藏

隐藏元素的提示条：

```
$('#element').tooltip('hide')
```

### 4．开关

开关元素的提示条：

```
$('#element').tooltip('toggle')
```

### 5．销毁

隐藏并销毁元素的提示条：

```
$('#element').tooltip('destroy')
```

## 4.8 弹出层

弹出层（见图 4-6）与提示条非常相似，不仅能显示内容，还可以显示标题。把光标悬停到元素上，就可以激活弹出层。弹出层的内容可以使用 Bootstrap Data API 来填充，而且要求元素本身包含提示条。弹出层的位置通过 `data-placement` 属性来指定，包括 `top`、`right`、`bottom` 和 `left`。

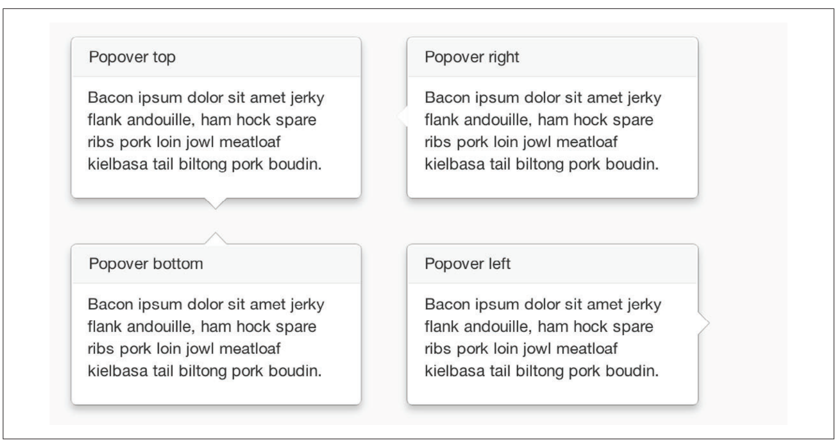



图 4-6：弹出层的位置

以下是创建弹出层的代码：

```
<a href="#" class="btn" rel="popover" title="Using Popover"
data-content="Just add content to the data-content attribute.">Click Me!</a>
```

## 4.8.1 用法

要通过 JavaScript 打开弹出层，使用 `.popover()` 函数，传入需要的选项：

```
$('#example').popover(options)
```

## 4.8.2 选项

所有选项都可以通过 Bootstrap Data API 设置，当然也可以通过 JavaScript 设置（见表 4-7）。

表4-7 弹出层的选项

| 名 称 | 类 型 | 默认值 | 说 明 |
|---|---|---|---|
| animation | Boolean | `true` | 为弹出层应用 CSS 的淡入淡出过渡 |
| html | Boolean | `false` | 弹出层支持插入 HTML。如果值为 `false`，则会使用 jQuery 的 `text` 方法。如果担心 XSS 攻击，那就使用文本吧 |
| placement | String \| Function | `'top'` | 设置弹出层的位置：`top` \| `bottom` \| `left` \| `right` |
| selector | String | `false` | 如果提供了选择符，则将弹出层委托给指定的目标 |
| trigger | String | `'hover focus'` | 设置如何触发弹出层：`click` \| `hover` \| `focus` \| `manual`。可以传入多个触发方式，以空格分隔即可 |
| title | String \| Function | `''` | 如果 `title` 属性不存在，则使用这个默认值 |
| content | String \| Function | `''` | 如果 `data-content` 属性不存在，则使用这个默认值 |
| delay | Number \| Object | `0` | 延迟显示和隐藏弹出层的时间（毫秒）——手动触发弹出层时无效<br>如果只指定一个数值，则隐藏和显示都使用该数值<br>对象的结构为 `delay: { show: 500, hide: 100 }` |
| container | String | `false` | 将弹出层附加到特定元素，如 `container: 'body'` |

### 4.8.3 方法

以下是与弹出层相关的方法。

#### 1. 选项

给一组元素添加弹出层：

```
$().popover(options)
```

#### 2. 显示

显示元素的弹出层：

```
$('#element').popover('show')
```

#### 3. 隐藏

隐藏元素的弹出层：

```
$('#element').popover('hide')
```

#### 4. 开关

开关元素的弹出层：

```
$('#element').popover('toggle')
```

#### 5. 销毁

隐藏并销毁元素的弹出层：

```
$('#element').popover('destroy')
```

## 4.9 警示框

使用 Data API 很容易给警示框添加关闭功能（见图 4-7）。




图 4-7：错误警示消息

## 4.9.1 用法

使用 JavaScript 调用 `.alert()` 方法，或者通过在链接或按钮中添加数据属性都可以让警示框具有关闭功能。

在 JavaScript 中要这样做：

```
$(".alert").alert()
```

通过 Data API 要这样做：

```
<a class="close" data-dismiss="alert" href="#">&times;</a>
```

## 4.9.2 关闭方法

要让所有警示框都具有关闭功能，传入如下选项。要让警示框以动画方式关闭，则要给它们应用 `.fade` 和 `.in` 类。

```
$(".alert").alert('close')
```

## 4.9.3 事件

警示框有两个事件，如表 4-8 所示。

表4-8　警示框相关的类

| 事　　件 | 说　　明 |
|---|---|
| close | 这个事件在实例方法 `close` 被调用后立即触发 |
| closed | 这个事件在警示框被关闭（CSS 过渡完成）后触发 |

例如，要在警示框关闭后执行操作，可以这样写代码：

```
$('#my-alert').bind('closed', function () {
  // 执行某些操作……
})
```

# 4.10　按钮

第 3 章介绍过按钮。在 Bootstrap 中，可以让按钮像链接，或者像表单中的按钮。通过这个插件可以为按钮赋予交互功能，比如显示加载状态或组成工具条似的按钮组。

## 4.10.1　加载状态

要让按钮显示加载状态（见图 4-8），只要给按钮添加 `data-loading-text="Loading..."` 属性即可：

```
<button type="button" class="btn btn-primary" data-loading-text="Loading...">
Submit!</button>
```

按钮被单击后，会被添加 `.disabled` 类，变灰，不再响应单击。

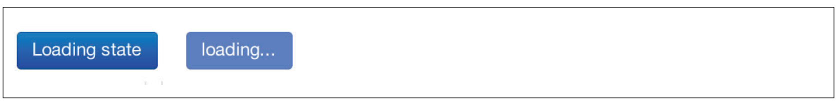



图 4-8：加载按钮

## 4.10.2　单选开关

单击带有 `data-toggle="button"` 属性的按钮（见图 4-9），会给它添加 `.active` 类：

```
<button type="button" class="btn btn-primary" data-toggle="button">Toggle
</button>
```

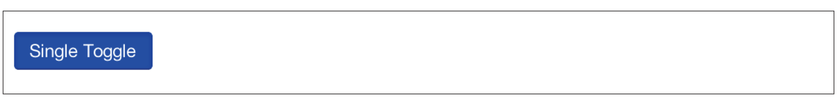



图 4-9：开关按钮

## 4.10.3　复选按钮

按钮也可以像复选框一样（见图 4-10），让用户从按钮组中选择多个按钮。要添加这个功能，给 `.btn-group` 添加 `data-toggle= "buttons-checkbox"` 属性：

```
<div class="btn-group" data-toggle="buttons-checkbox">
  <button type="button" class="btn btn-primary">Left</button>
  <button type="button" class="btn btn-primary">Middle</button>
  <button type="button" class="btn btn-primary">Right</button>
</div>
```




图 4-10：复选按钮

## 4.10.4　单选按钮

单选按钮（见图 4-11）与复选按钮的区别就是不允许选择多个，即每次只能选择按钮组中的一个。要在按钮组中添加单选按钮，需要给 `.btn-group` 添加 `data-toggle="buttons-radio"` 属性：

```
<div class="btn-group" data-toggle="buttons-radio">
  <button type="button" class="btn btn-primary">Left</button>
  <button type="button" class="btn btn-primary">Middle</button>
  <button type="button" class="btn btn-primary">Right</button>
</div>
```

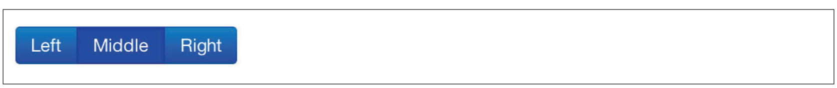



图 4-11：单选按钮

## 4.10.5　用法

可以对任何类和 ID 调用 `.button()` 方法。比如，要通过 JavaScript 启用 `.nav-tabs` 中的按钮，可以运行以下代码：

```
$('.nav-tabs').button()
```

### 4.10.6　方法

以下是与按钮相关的方法调用。

#### 1．开关

切换状态，让人感觉按钮被按下了：

```
$().button('toggle')
```

#### 2．加载

加载过程中，按钮是禁用的，显示的文本会变成 `data-loading-text` 属性的值：

```
<button type="button" class="btn" data-loading-text="loading stuff..." >...
</button>
```

#### 3．重置

重置按钮状态，显示原来的文本。适合让按钮显示最初状态的情况：

```
$().button('reset')
```

#### 4．字符串

传入这个参数可以让按钮显示字符串：

```
$().button('string')
```

要重置按钮状态，显示新内容，可以这样：

```
<button type="button" class="btn" data-complete-text="finished!" >...</button>

<script>
  $('.btn').button('complete')
</script>
```

## 4.11　折叠框

这个插件简化了折叠框的创建（见图 4-12）。无论是创建折叠式导航还是内容区，都可以通过折叠框的选项进行设置。

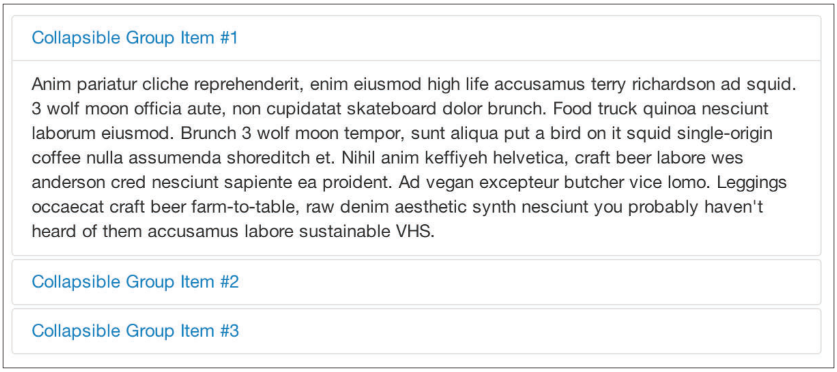



图 4-12：折叠框

以下代码用于创建可折叠的区域：

```
<div class="accordion" id="accordion2">
  <div class="accordion-group">
    <div class="accordion-heading">
      <a class="accordion-toggle" data-toggle="collapse" data-parent=
          "#accordion2" href="#collapseOne">
        Collapsible Group Item #1
      </a>
    </div>
    <div id="collapseOne" class="accordion-body collapse in">
      <div class="accordion-inner">
        Anim pariatur cliche...
      </div>
    </div>
  </div>
  <div class="accordion-group">
    <div class="accordion-heading">
      <a class="accordion-toggle" data-toggle="collapse" data-parent=
          "#accordion2" href="#collapseTwo">
        Collapsible Group Item #2
      </a>
    </div>
    <div id="collapseTwo" class="accordion-body collapse">
      <div class="accordion-inner">
        Anim pariatur cliche...
      </div>
    </div>
  </div>
</div>
...
```

可以使用数据属性让所有内容都可以折叠：

```
<button type="button" class="btn btn-danger" data-toggle="collapse"
data-target="#demo">
  simple collapsible
</button>

<div id="demo" class="collapse in"> ... </div>
```

## 4.11.1 用法

### 1. 数据属性方式

与使用 Data API 的其他插件一样，一切都可以在标记中搞定。给一个元素添加 `data-toggle="collapse"` 和 `data-target` 属性，就能令其控制相应的可折叠元素。其中，`data-target` 属性的值可以是表示可折叠元素的 CSS 选择符。另外，要给可折叠元素添加 `.collapse` 类。如果默认想打开哪个元素，就再给它添加一个 `.in` 类。

要指定类似折叠组的元素，为每个可折叠元素添加 `data-parent="#selector"`。

### 2. JavaScript方式

通过下面的 JavaScript 方法也可以激活可折叠插件：

```
$(".collapse").collapse()
```

## 4.11.2 选项

表 4-9 列出了折叠框相关的选项，可以通过数据属性或 JavaScript 来设置这些选项。

表4-9　折叠框选项

| 名　称 | 类　型 | 默认值 | 说　　明 |
| --- | --- | --- | --- |
| parent | 选择符 | `false` | 如果指定了选择符，则所有位于这个指定的父元素之下的可折叠元素都会在父元素可见时折叠起来（类似传统的折叠框） |
| toggle | Boolean | `true` | 切换可折叠元素的状态 |

### 4.11.3　方法

以下是与可折叠元素相关的方法调用。

#### 1. 选项

把元素激活为可折叠元素，接受可选的选项对象：

```
.collapse(options)
```

#### 2. 开关

切换可折叠元素的状态：

```
$('#myCollapsible').collapse({
  toggle: false
})
.collapse('toggle')
```

#### 3. 显示

显示可折叠元素：

```
.collapse('show')
```

#### 4. 隐藏

隐藏可折叠元素：

```
.collapse('hide')
```

### 4.11.4　事件

可折叠元素相关的事件有 4 个，见表 4-10。

表4-10　折叠框相关事件

| 事　　件 | 说　　明 |
|---|---|
| show | 在实例方法 show 被调用后触发 |
| shown | 在可折叠元素对用户完全可见（CSS 过渡完成）后触发 |
| hide | 在实例方法 hide 被调用后触发 |
| hidden | 在可折叠元素对用户完全不可见（CSS 过渡完成）后触发 |

在 <div> 折叠之后，可以使用以下代码执行其他操作：

```
$('#myCollapsible').on('hidden', function () {
  // 执行其他操作……
})
```

## 4.12　传送带

Bootstrap 的传送带（见图 4-13）是一个灵活的、响应式的幻灯片切换插件。除可以具有响应性，其内容也可以是图片、子框架、视频，或者其他任何内容。




图 4-13：传送带

以下是实现 Bootstrap 传送带的代码：

```
<div id="myCarousel" class="carousel slide">
  <!-- Carousel items -->
  <div class="carousel-inner">
    <div class="active item">...</div>
    <div class="item">...</div>
    <div class="item">...</div>
  </div>
  <!-- Carousel nav -->
```

```
    <a class="carousel-control left" href="#myCarousel" data-slide="prev">
        &lsaquo;</a>
    <a class="carousel-control right" href="#myCarousel" data-slide="next">
        &rsaquo;</a>
</div>
```

## 4.12.1 用法

要实现传送带效果，需要添加上面所示的标记，不用数据属性，只要添加相应的类即可。在 JavaScript 中，可以像下面这样手动调用传送带插件：

```
$('.carousel').carousel()
```

## 4.12.2 选项

选项可以通过数据属性设置，也可以通过 JavaScript 传入。表 4-11 列出了传送带相关的插件。

表4-11 传送带选项

| 名 称 | 类 型 | 默认值 | 说 明 |
|---|---|---|---|
| interval | number | 5000 | 每次自动切换间隔的时间，如果未指定则使用默认时间 |
| pause | string | "hover" | 在 `mouseenter` 时暂停轮播，在 `mouseleave` 时恢复 |

## 4.12.3 方法

以下是与传送带插件相关的方法调用。

### 1. 选项

以可选的选项对象初始化并启动传送带：

```
$('.carousel').carousel({
  interval: 2000
})
```

### 2．轮播

从左到右轮播传送带中的项：

```
.carousel('cycle')
```

### 3．暂停

暂停传送带的轮播：

```
.carousel('pause')
```

### 4．数值

把传送带切换到特定的项（类似数组，基于 0 计数）：

```
.carousel('number')
```

### 5．前一项

切换到前一项：

```
.carousel('prev')
```

### 6．后一项

切换到后一项：

```
.carousel('next')
```

## 4.12.4 事件

表 4-12 列出了与传送带相关的事件。

表4-12 传送带相关的事件

| 事　件 | 说　明 |
|---|---|
| slide | 在实例方法 `slide` 被调用后触发 |
| slid | 在传送带完成切换动画之后触发 |

# 4.13 预先输入

用这个插件可以创建带预先输入的表单（见图 4-14）。举个例子，可以在“州”这个字段中预先加载所有州名，或者使用 JavaScript 通过 AJAX 调用来搜索结果。




图 4-14：预先输入

## 4.13.1 用法

使用 Data API，可以通过 `data-source` 属性添加预先输入的内容。内容的格式可以是 JSON 数组，也可以来自函数：

```
<input
        type="text"
        class="span3"
        data-provide="typeahead"
        data-items="4"
        data-source="[
                'Alabama',
                'Alaska',
                'Arizona',
                'Arkansas',
                'California',
                ...
                ]"
>
```

要通过 JavaScript 直接调用，使用如下代码：

```
$('.typeahead').typeahead()
```

## 4.13.2 选项

表 4-13 列出了与预先输入相关的选项。

表4-13 预先输入相关的选项

| 名 称 | 类 型 | 默认值 | 说 明 |
|---|---|---|---|
| source | Array, Function | [ ] | 要查询的数据源。可以是字符串数组，也可以是一个函数。如果是函数，则接受两个参数，一个是输入字段中要查询的值 `query`，另一个是回调函数 `process`。函数可以直接返回数据源，实现同步查询，也可以通过 `process` 回调函数进行异步查询 |
| items | Number | 8 | 下拉选项列表中最多显示几项 |
| minLength | Number | 1 | 触发自动完成建议最少需要几个字符 |
| matcher | Function | 不区分大小写 | 用于检测查询是否找到了匹配项，接受一个参数 `item`，用于对比查询。要访问当前查询，使用 `this.query`。如果找到匹配项，返回 `true` |
| sorter | Function | 完全匹配<br>区分大小写<br>不区分大小写 | 用于排序自动完成建议的方法，接受一个参数 `items`，作用域为当前的预先输入元素。要访问当前查询，使用 `this.query` |
| updater | Function | 返回选中项 | 用于返回选中项的方法，接受一个参数 `item`，作用域为当前的预先输入元素 |
| highlighter | Function | 突出显示所有默认匹配项 | 用于突出显示自动完成结果的方法，接受一个参数 `item`，作用域为当前的预先输入元素。应该返回 HTML |

# 4.14 粘条

粘条（affix）插件会让一个 `<div>` 粘附在页面上某个位置。开始时，粘条出现在页面的某个位置，但当页面滚动一定距离时，它就固定不动了，无论页面如何滚动它都是静止的。

## 4.14.1 用法

要实现粘条插件，可以使用数据属性，也可以使用 JavaScript。记住，必须定位相应的元素，以便它能粘附在页面上。定位是通过 `data-spy` 属性控制的，这个属性的值可以是：`affix`、`affix-top` 或 `affix-bottom`。然后，再通过 `data-offset` 来设置滚动距离。

```
<div data-spy="affix" data-offset-top="200">
        ...
</div>
```

## 4.14.2 选项

表 4-14 列出了粘条相关的选项。

表4-14　粘条相关的选项

| 名　称 | 类　型 | 默认值 | 说　　明 |
| --- | --- | --- | --- |
| offset | Number \| Function \| Object | 10 | 计算滚动距离时偏移屏幕的像素数。如果只提供了一个数值，则将该数值同时应用给上方和左侧。要检测某一个方向，需要提供一个对象，如：`offset: { x: 10 }`。在需要动态设置偏移量时，可以提供一个函数（适合响应式设计） |

第 5 章

# 实际使用Bootstrap

## 5.1 GitHub项目

与众多优秀开源项目一样，Bootstrap 也不仅仅是几位核心开发人员的成果，而是整个开发社区的功劳。GitHub 是托管项目的一个代码库，在本书写作时，Bootstrap 是关注度最高的项目。超过 50000 颗星星，16000 多个分支，充分表明这个项目有多么活跃。正如我们在第 1 章提到的，要使用 Bootstrap，可以在它的站点上下载，也可以使用 Git 下载。

Git 是一个自由的开源版本控制系统。包括 Bootstrap 在内的大量项目都使用 GitHub 做项目管理，而 GitHub 就是基于 Git 的一个在线代码库。要用 Git 下载 Bootstrap，在命令行执行如下命令：

```
$ git clone https://github.com/twitter/bootstrap.git
```

这样克隆 Bootstrap 将得到所有文件，不仅仅是 CSS/JavaScript，还有文档和针对动态 JavaScript 元素的 LESS 文件。

如果你愿意，可以使用 LESS 文件编译自己的 Bootstrap 版本，实现对功能的定制。

## 5.2 定制Bootstrap

可以下载全部源代码，也可以从颜色、大小或插件等几个方面进行定制，这些都可以通过 Bootstrap 网站做到。

在定制 Bootstrap 的页面上（见图 5-1），可以选择包含哪些组件。比如，可以不选择响应性功能，或者不选择与既有按钮冲突的按钮相关的类。当然，jQuery 插件也是可选的。假如你知道自己不需要模态框或传送带插件，那就不要包含它们，这样下载的文件会更小。

最后，还可以配置 LESS 变量。包括列数、颜色，等等都可以修改。经过一番定制，你一定可以得到适合自己需要的版本。

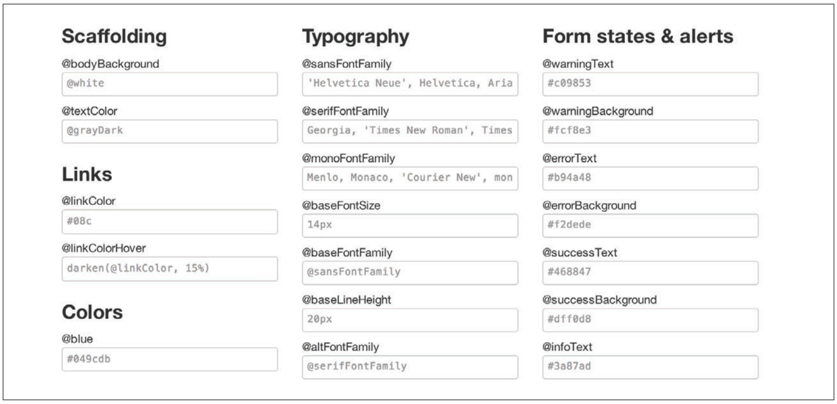



图 5-1：定制 Bootstrap

## 5.2.1　使用LESS

在 Bootstrap 中使用 LESS 有几种不同的方式（见图 5-2）。第一种最简单，就是使用 CodeKit（见图 5-3）或 SimpLESS 之类的预处理器。利用这些工具可以查看特定的文件和文件夹。在保存某些文件时，它们会帮你构建主 CSS 文件。除了使用传统的 CSS 技术，还可以使用混入、函数等高级功能，以及通过修改某些变量达到改变站点外观的目的。

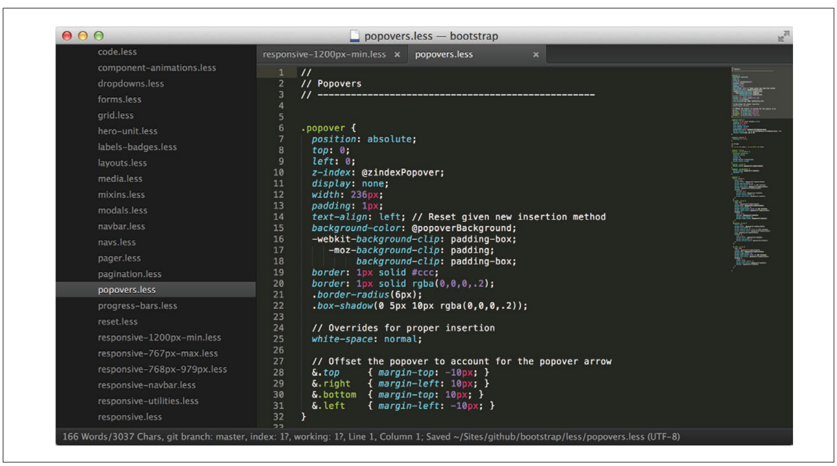



图 5-2：通过 LESS 定制 Bootstrap

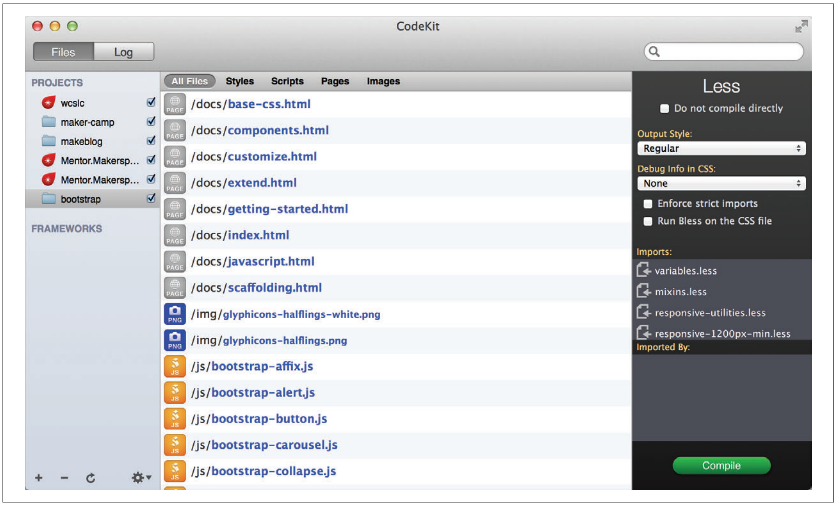



图 5-3：用 CodeKit 查看 Bootstrap 代码

LESS 是用于编写 CSS 的一种动态样式表语言，它支持变量、函数和混入功能。Bootstrap 的 /less/ 文件夹里包含一些 LESS 文件。如果想设置全局变量，需要关注 variables.less 和 mixins.less。

在 variables.less 中，可以找到 Bootstrap 的所有全局变量。以修改所有链接颜色为例，只要修改 LESS 变量，重新编译，就可以了。

```
/* 修改前 */
@linkColor:                #08c;
@linkColorHover:       darken(@linkColor, 15%);

/* 修改后 */
@linkColor:                #7d00cc;
@linkColorHover:       darken(@linkColor, 15%);
```

这样，所有链接就会变成紫色，网站中所有按钮及其他引用 `@linkColor` 的界面元素都会得到更新。

一开始的时候，我很犹豫要不要使用 LESS。毕竟，我写 CSS 都很长时间了，我不想改变。可 LESS 的结构真的能节省大量时间，何乐不为呢？

## 5.2.2 Snippets插件

要想快速地基于 Bootstrap 写个网站，建议使用 Sublime Text 2 和 Bootstrap Snippets 插件（见图 5-4），后者下载地址为：https://github.com/devtellect/sublime-twitter-bootstrap-snippets/，来自 DEVtellect 。这个插件可以让你使用快捷方式添加任意组件。

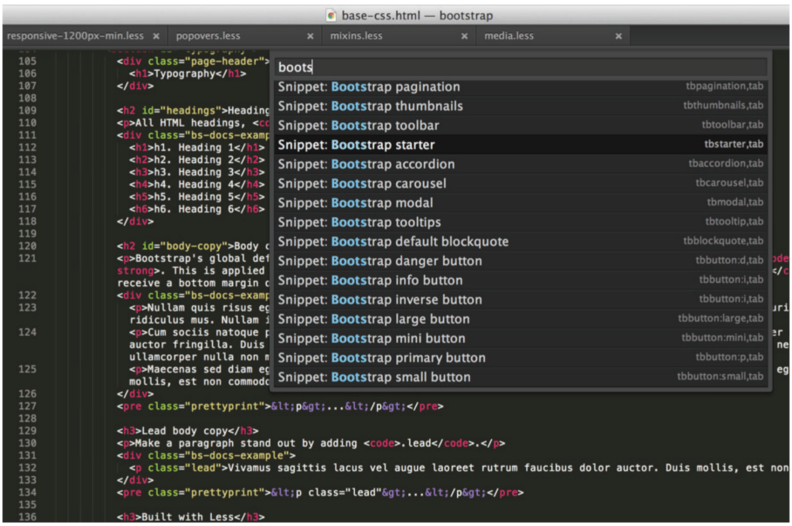



图 5-4：Bootstrap Snippets 插件

要安装这个插件，需要把其代码克隆到你的 packages 文件夹里：

```
git clone git@github.com:devtellect/sublime-twitter-bootstrap-snippets.git
```

不少流行的文本编辑器都有这种插入文本片段的功能。

## 5.2.3 Photoshop模板

除了文本片段，还有一些针对 Bootstrap 元素设计的 Photoshop 文件。我比较喜欢 Repix Design 的设计（http://gui.repixdesign.com/#bootstrap），图 5-5 展示了它们的网站。

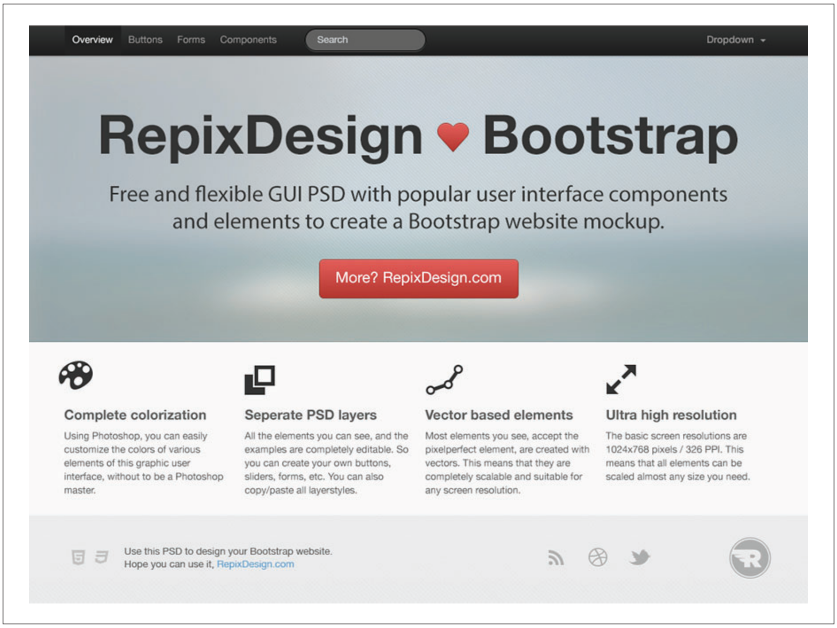



图 5-5：Repix design

这个网站中的 Photoshop 模板具有如下特点：

- 可调整的颜色
- 独立的图层
- 基于矢量

模板 PSD 文件是免费的，但要求使用者在 Twitter 上转发。这种免费转发的模式在开源编程领域渐成气候。

### 5.2.4 主题

如果你安装了简单的 Bootstrap，但后来想添加一些装饰，我向你推荐两个 Bootstrap 主题网站，其中的主题有的免费，有的收费：一个是 {Wrap}Bootstrap（https://wrapbootstrap.com/），另一个是 Bootswatch（http://bootswatch.com/）。这些主题都有很多选项，通过它们一定能把你的网站设计得新颖别致。

## 5.3　获得灵感

如果你想得到更多灵感，可以看一看 Built With Bootstrap（http://builtwithbootstrap.com/），一个由用户提交 Bootstrap 站点截图的轻博客。浏览一下那些使用 Bootstrap 的网站是很有意思的。

## 5.4　小结

显而易见，Bootstrap 几乎可以适用于任何网站。内置响应式框架、丰富的 JavaScript 插件、健壮的界面组件，这些为快速创建功能强大的网站提供了便利。过去一年使用 Bootstrap 的经历让我感觉非常好，希望这个项目将来会有更大发展。好样的，太棒了，我要感谢 Jacob Thornton 和 Mark Otto 发起并创建了这个万能无敌的项目。

# 作者及封面介绍

## 作者简介

Jake Spurlock 是 O'Reilly Media 的开发者，主要开发 MAKE。MAKE 作为一个 DIY 杂志，发起了 Maker Faire（制汇节），该活动旨在教会人们动手制作，让世界变得更美好。Jake 是犹他州人，一年半前搬到了加州葡萄酒之乡。

## 封面介绍

本书封面动物是芬兰马。芬兰马是芬兰国马，由于这种马能耕种、驾车，还可供人驱驰、消遣，而且具有很高的经济价值，因此世人美其名曰“万能马”。虽然这种马在芬兰已经繁殖、演化了几个世纪，但它们的起源至今仍然是个谜。

芬兰马大部分是栗褐色，间杂极少量黑色、银色或枣红色。1800 年代，栗色芬兰马大约只占一半，但经过 19 世纪和 20 世纪不断地优选，多数其他基因都被淘汰了。人们都认为栗色是这种马的正宗颜色，芬兰马国家繁育协会也致力于消除“非正宗”颜色。另外，大约 155 厘米的平均身高也是人工着意选择的，因为这个高度适合体力劳动，也可用于骑行和娱乐。

今天，大多数芬兰马都用于比赛以及其他消遣用途。20 世纪中叶，芬兰马一度是赛马场上唯一的马种。但此后，其他马也加入了赛马阵营。这种马也被很多骑马学校使用，或者用于马疗法。

# Bootstrap用户手册：设计响应式网站

这本书教你怎么用Bootstrap框架轻松设计出“杀手级”界面及响应式网站。从怎么用Bootstrap的HTML/CSS工具和现成模板构建页面开始，逐步深入地掌握它内置的交互组件和jQuery插件，常常是一行代码都不用写。

作为源自Twitter的一个开源框架，而且可以从GitHub上自由下载，Bootstrap推崇“移动优先”（Mobile First）的设计理念，还支持动态调整网页布局、创建响应式网站。只要稍微懂一些HTML、CSS和JavaScript，就可以创建出在桌面显示器、智能手机和平板电脑中看起来同样完美的网站或应用。这本书会告诉你怎么做。

- 熟悉Bootstrap的文件结构、网格系统和容器布局；
- 学习HTML中用于排版、代码、表格、表单、按钮、图片以及图标的元素；
- 设计界面及其他网页元素，比如导航、面包屑，以及自定义模态窗口等；
- 使用jQuery插件实现渐进的幻灯片、标签页式界面和下拉菜单；
- 在LESS文件中修改布局栏数和字体颜色。

“……Jake Spurlock是一位前端工程师，已经用Bootstrap创建了几个网站……他认为是我的那几篇博文引领他结识了强大优雅的Bootstrap。现在，我把你交给他，让他为你展示Bootstrap的魔法，我们一块推动Web开发向前进。”

——Dave Winer

*Scripting News*杂志编辑

Jake Spurlock是MakerMedia的开发者，*MAKE*杂志的制作人，他发起了全球各地的“制汇节”活动，该活动旨在教会人们动手制作，让世界变得更美好。Jake是犹他州人，一年半前搬到了加州葡萄酒之乡。

封面设计：Karen Montgomery　张健
图灵社区：www.ituring.com.cn
新浪微博：@图灵教育 @图灵社区
反馈/投稿/推荐信箱：contact@turingbook.com
热线：(010)51095186转604

分类建议　计算机/网站设计

人民邮电出版社网址：www.ptpress.com.cn
O’Reilly Media, Inc.授权人民邮电出版社出版
此简体中文版仅限于中国大陆（不包含中国香港、澳门特别行政区和中国台湾地区）销售发行
This Authorized Edition for sale only in the territory of People's Republic of China (excluding Hong Kong, Macao and Taiwan)

O’REILLY®
oreilly.com.cn

ISBN 978-7-115-32735-2

ISBN 978-7-115-32735-2
定价：35.00元

欢迎加入

# 图灵社区

## 最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

## 最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版的梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要你有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

## 最直接的读者交流平台

在图灵社区，你可以十分方便地写作文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、审读、评选等多种活动，赢取积分和银子，积累个人声望。

ituring.com.cn